(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 390 051 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 90105762.0

(51) Int. Cl.5: G05D 1/08

(22) Date of filing: 27.03.90

(30) Priority: 31.03.89 US 332320

(43) Date of publication of application:
03.10.90 Bulletin 90/40

(84) Designated Contracting States:
DE FR GB NL SE

(71) Applicant: HONEYWELL INC.
Honeywell Plaza
Minneapolis MN 55408(US)

(72) Inventor: Bhanu, Bir
1889 29th N.W.
New Brighton, MN 55112(US)
Inventor: Burger, Wilhelm
Hagenstrasse 61/5
A-4040 Linz(AT)

(74) Representative: Herzbach, Dieter et al
Honeywell Europe S.A. Holding KG Patent & License Dept. Kaiserleistrasse 55 Postfach 10 08 65
D-6050 Offenbach am Main(DE)

(54) Method and apparatus for computing the self-motion of moving imaging devices.

(57) Determining the self-motion in space of an imaging device (e.g., a television camera) by analyzing image sequences obtained through the device. Three-dimensional self-motion is expressed as a combination of rotations (ϕ,θ,ψ) about the horizontal and vertical camera axes and the direction of camera translation. The invention computes the rotational and translational components of the cameras self-motion exclusively from visual information. Robust performance is achieved by determining the direction of heading (i.e., the focus of expansion) as a connected region instead of a single location on the image plane. The method can be used to determine when the direction of heading is outside the current field of view and when there is zero or very small camera translation.

EP 0 390 051 A2

METHOD AND APPARATUS FOR COMPUTING THE SELF-MOTION OF MOVING IMAGING DEVICES

Field of the Invention

The present invention pertains to a method and apparatus for the determination of three-dimensional self-motion of moving imaging devices according to the preamble of the independent claims.

Background of the Invention

Visual information is an indispensable clue for the successful operation of an autonomous land vehicle. Even with the use of sophisticated inertial navigation systems, the accumulation of position error requires periodic corrections. Operation in unknown environments or mission tasks involving search, rescue, or manipulation critically depend upon visual feedback.

Assessment of scene dynamics becomes vital when moving objects may be encountered, e.g., when the autonomous land vehicle follows a convoy, approaches other vehicles, or has to detect moving threats. For the given case of a moving camera, such as one mounted on the autonomous land vehicle, image motion can supply important information about the spatial layout of the environment ("motion stereo") and the actual movements of the land vehicle.

Previous work in motion analysis has mainly concentrated on numerical approaches for the recovery of three-dimentional (3-D) motion and scene structure from two-dimensional (2-D) image sequences. the most common approach is to estimate 3-D structure and motion in one computational step by solving a system of linear or non-linear equations. This technique is characterized by several severe limitations. First, it is known for its notorious noise-sensitivity. To overcome this problem, some researchers have extended this technique to cover multiple frames. Secondly, it is designed to analyze the relative motion and 3-D structure of a single rigid object. To estimate the egomotion of an autonomous land vehicle (ALV), having the imaging device or camera, and the accompanying scene structure, the environment would have to be treated as a large rigid object. However, rigidness of the environment cannot be guaranteed due to the possible presence of moving objects in the scene. The consequence of accidentally including a moving 3-D point into the system of equations, representing the imaged environment, in the best case, would be a solution (in terms of motion and structure) exhibiting a large residual error, indicating some non-rigid behavior. The point in motion, however, could not be immediately identified from this solution alone. In the worst case (for some forms of motion), the system may converge towards a rigid solution (with small error) in spite of the actual movement in the point set. This again shows another (third) limitation: there is no suitable means of expressing the ambiguity and uncertainty inherent to dynamic scene analysis.

It is therefore the object of the present invention to devise a method and an apparatus with which the self-motion of an imaging device can be unambiguously evaluated. This object is achieved by the characterizing features of the independent claims. Advantageous embodiments of the inventive method and apparatus may be taken from the dependent claims.

The invention, that solves the aforementioned problems, is novel in two important aspects. The scene structure is not treated as a mere by-product of the motion computation but as a valuable means to overcome some of the ambiguities of dynamic scene analysis. The key idea is to use the description of the scene's 3-D structure as a link between motion analysis and other processes that deal with spatial perception, such as shape-from-occlusion, stereo, spatial reasoning, etc. A 3-D interpretation of a moving scene can only be correct if it is acceptable by all the processes involved.

Secondly, numerical techniques are largely replaced by a qualitative strategy of reasoning and modeling. Basically, instead of having a system of equations approaching a single rigid (but possibly incorrect) numerical solution, multiple qualitative interpretations of the scene are maintained. All the presently existing interpretations are kept consistent with the observations made in the past. The main advantage of this approach of the present invention is that a new interpretation can be supplied immediately when the currently favored interpretation turns out to be unplausible.

The problem of determining the motion parameters of a moving camera relative to its environment from a sequence of images is important for applications for computer vision in mobile robots. Short-term control, such as steering and braking, navigation, and obstacle detection/avoidance are all tasks that can effectively utilize this information.

Summary of the Invention

The present invention deals with the computation of sensor platform motion from a set of displacement vectors obtained from consecutive pairs of images. It is directed for application to autonomous robots and land vehicles. The effects of camera rotation and translation upon the observed image are overcome. The new concept of "fuzzy" focus of expansion (FOE), which marks the direction of vehicle heading (and provides sensor rotation), is exploited. It is shown that a robust performance for FOE location can be achieved by computing a 2-D region of possible FOE-locations (termed "Fuzzy FOE") instead of looking for a single-point FOE. The shape of this FOE is an explicit indicator of the accuracy of the result. Given the fuzzy FOE, a number of very effective inferences about the 3-D scene structure and motion are possible and the fuzzy FOE can be employed as a practical tool in dynamic scene analysis. The results are realized in real motion sequences.

The problem of understanding scene dynamics is to find consistent and plausible 3-D interpretations for any change observed in the 2-D image sequence. Due to the motion of the autonomous land vehicle (ALV), containing the scene sensing device, stationary objects in the scene generally do not appear stationary in the image, whereas moving objects are not necessarily seen in motion. The three main tasks of the present approach for target motion detection and tracking are: 1) to estimate the vehicle's motion; 2) to derive the 3-D structure of the stationary environment; and 3) to detect and classify the motion of individual targets in the scene. These three tasks are interdependent. The direction of heading (i.e., translation) and rotation of the vehicle are estimated with respect to stationary locations in the scene. The focus of expansion (FOE) is not determined as a particular image location, but as a region of possible FOE-locations called the fuzzy FOE. We present a qualitative strategy of reasoning and modeling for the perception of 3-D space from motion information. Instead of refining a single quantitative description of the observed environment over time, multiple qualitative interpretations are maintained simultaneously. This offers superior robustness and flexibility over traditional numerical techniques which are often ill-conditioned and noise-sensitive. A rule-based implementation of this approach is discussed and results on real ALV imagery are presented.

The system of the present invention tracks stationary parts of the visual environment in the image plane, using corner points, contour segments, region boundaries and other two-dimensional tokens as references. This results in a set of 2-D displacement vectors for the selected tokens for each consecutive pair of camera images. The self-motion of the camera is modeled as two separate rotations about horizontal and the vertical axes passing through the lens center and a translation in 3-D space. If the camera performs pure translation along a straight line in 3-D space, then (theoretically) all the displacement vectors extend through one particular location in the image plane, called the focus of expansion (FOE) under forward translation or focus of contraction (FOC) under backward translation. The 3-D vector passing through the lens center and the FOE (on the image plane) corresponds to the direction of camera translation in 3-D space.

The invention can provide the directions of instantaneous heading of the vehicle (FOE) within $1^{\bullet}$ and self motions of moving imaging devices can be accurately obtained. This includes rotations of $\pm 5^{\bullet}$ or larger in horizontal and vertical directions. To cope with the problems of noise and errors in the displacement field, a region of possible FOE-locations (i.e., the fuzzy FOE) is determined instead of a single FOE.

In practice, however, imaging noise, spatial discretization errors, etc., make it impractical to determine the FOE as an infinitesimal image location. Consequently, a central strategy of our method is to compute a region of possible FOE-locations (instead of a single position) which produces more robust and reliable results than previous approaches.

Brief Description of the Drawings

Figure 1 is a functional block diagram of the present invention.

Figure 2 is a diagram showing an extended application of the invention to three-dimensional scene constructing.

Figure 3 shows a camera model and corresponding coordinate system.

Figure 4 illustrates a successive application of horizontal and vertical rotation of the camera.

Figure 5 diagrams the effect of a focus of expansion location (FOE) for pure camera translation.

Figure 6 illustrates the concept of the FOE for discrete time steps during the motion of a vehicle having the camera.

Figure 7 shows the amount of expansion from the FOE for discrete time steps.

Figures 8a and 8b display a displacement field caused by horizontal and vertical rotation and translation of the camera and a derotated displacement field, respectively.

Figures 9a and 9b show an image plane and a rotation space, respectively.

Figures 10a and 10b show mappings of a polygon from rotation space to an image plane and vice versa, respectively.

Figures 11a-c illustrate a changing rotation polygon.

Figure 12 illustrates an intersection of displacement vectors with a vertical line which, if moved, changes the variance of intersection.

Figure 13 shows a displacement field used to evaluate various error functions.

Figures 14a-d reveal the standard deviation of intersection at a vertical cross section at position x for different amounts of vertical rotation.

Figures 15a-d reveal the standard deviation of intersection (square root) at a vertical cross section at position x for different amounts of vertical rotation with no horizontal rotation and with no pixel noise applied to the image locations.

Figures 16a-d reveal the standard deviation of intersection (square root) at a vertical cross section at position x for different amounts of vertical rotation with no horizontal rotation and with ±1 pixels of noise applied to the image locations.

Figures 17a-d reveal the standard deviation of intersection (square root) at a vertical cross section at position x for different amounts of vertical rotation with no horizontal rotation and with ±2 pixels of noise applied to the image locations.

Figures 18a-d show the location of minimum intersection standard deviation under varying horizontal rotation with the horizontal location of the FOE marked xf.

Figures 19a-d show the amount of minimum intersection standard deviation under varying horizontal rotation and with no noise added to the image locations.

Figures 20a-d show the amount of minimum intersection deviation under varying horizontal rotation and with ±2 pixels noise added to the image locations.

Figure 21 illustrates intersecting displacement vectors with two vertical lines which lie on the same side of the FOE.

Figures 22a-d show the correlation coefficient for the intersection of displacement vectors at two vertical lines under varying horizontal rotations with no noise added.

Figures 23a-d show the correlation coefficient for the intersection of displacement vectors at two vertical lines under varying horizontal rotations and with ±2 pixels of noise added to the image locations.

Figure 24 illustrates displacement vectors and measurement of their error.

Figure 25 shows how to determine the optimum two-dimensional shift for a set of displacement vectors.

Figure 26 reveals how FOE locations are dismissed if the displacement field resulting from the application of the optimal shift results in a vector not pointing away from the FOE.

Figures 27a-d illustrate the displacement field and minimum error at selected FOE locations.

Figures 28a-e display the effects of increasing the average length of displacement vectors upon the shape of the error function.

Figures 29a-e display the effects of increasing residual rotation in a horizontal direction upon the shape of the error function for relatively short vectors.

Figures 30a-e display the effects of increasing residual rotation in a vertical direction upon the shape of the error function for relatively short vectors.

Figures 31a-e show the effects of increasing residual rotation in horizontal and vertical directions upon the shape of the error function for relatively short vectors.

Figures 31f-j indicate the amount of optimal linear shift obtained under the same conditions in Figures 31a-e.

Figures 32a-e show the effects of the uniform noise applied to image point coordinates for a constant average vector length.

Figures 33a and 33b reveal the different effects of uniform noise applied to image point coordinates for shorter and longer average vector lengths, respectively.

Figure 34 reveals a side view of a camera traveling parallel to a flat surface.

Figures 35a-i show an original image sequence taken from a moving vehicle after edge detection and point detection with the selected points located at the lower-left corners of their marks.

Figures 35j-p show the original image sequence after edge detection and point selection.

Figures 36a-p illustrate the displacement vectors and estimates of vehicle motion for the image

sequence shown in Figures 35a-p.

Figure 37 illustrates a specific hardware implementation of the embodiment of the invention.

Description of the Present Embodiment

Figures 1 and 2 reveal the main portions of the present invention. Figure 2 also expands on 3-D interpretations of 2-D images in items 118 and 120. Item 116 incorporates portions of items 122 and 124 of Figure 1. First, significant features (points, boundaries, corners, etc.) are extracted by feature extraction and tracking 114 from the image data 112 and the 2-D displacement vectors are computed for this set of features. For the examples shown here, points were selected and tracked between individual frames by item 114. Automatic techniques suitable for this task are in the related art. In the second step, the vehicle's direction of translation, i.e. the focus of expansion (FOE), and the amount of rotation in space are determined. The effects of vehicle motion on the FOE computation is described below. Nearly all the necessary numerical computation is performed in the FOE computation stage, which also is described below. The third step (at 2-D change analysis 118) constructs an internal 3-D model of the scene. Also disclosed are the concepts and operation of the qualitative scene model 120. Experiments with the present invention on real imagery taken from a moving ALV are discussed below.

System 110 contains the following three main components --token tracking 114, FOE seeker 124 and optimal derotation 122-- in Figure 1. The 2-D displacement vectors for selected image tokens are determined in the first stage (token tracking 114). Since those original displacement vectors are caused by some arbitrary and (at this point) unknown camera motion, including camera rotations, they do not yet exhibit the characteristic radial pattern of pure camera translation.

The second component (FOE seeker 124) selects a set of candidate locations for the FOE and forms a connected image region of feasible FOE-locations plus the range of corresponding camera rotations, based on the results from the third component (optimal derotation 122). A particular FOE-location is feasible, if the corresponding error value (computed by the optimal derotation module 122) is below some dynamically adjusted threshold. The size of the final FOE-region reflects the amount of uncertainty contained in the visual information (large regions reflect high uncertainty).

The third component (optimal derotation 122) determines the optimal 3-D camera rotations for a particular (hypothesized) FOE-location. This is accomplished by simulating the effects of reverse camera rotations upon the given set of displacement vectors. The camera is virtually rotated until the modified displacement field is closest to a radial expansion pattern with respect to the selected FOE-location. Module 122 returns the necessary amount of reverse rotation and the deviation from a radial displacement field (i.e., an error value).

Component 123, comprising FOE seeker 124 and optimal derotation 122, represents the fuzzy FOE means which outputs the region of possible FOE locations.

The first step of the present invention is to estimate the vehicle's motion relative to the stationary environment using visual information. Arbitrary movement of an object in 3-D space and thus the movement of the vehicle itself can be described as a combination of translation and rotation. While knowledge about the composite vehicle motion is essential for control purposes, only translation can supply information about the spatial layout of the 3-D scene (motion stereo). This, however, requires the removal of all image effects resulting from vehicle rotation. For this purpose, we discuss the changes upon the image that are caused by individual application of the "pure" motion components.

It is well-known that any rigid motion of an object in space between two points in time can be decomposed into a combination of translation and rotation. While many researchers have used a velocity-based formulation of the problem, the following treatment views motion in discrete time steps.

The viewing geometry involves a world coordinate system, (XYZ) as illustrated in Figure 3. Figure 3 shows the camera-centered coordinate system 130, lens center 126, image plane 128, and angles ϕ, θ and ψ of rotation. The origin O of coordinate system 130 is located at lens center 126. Focal length f is the distance between lens center 126 and image plane 128. Each 3-D point (XYZ) may be mapped onto image location (X,Y). Angles ϕ, θ and ψ specify angles of camera rotation about the X, Y and Z axes, respectively. Given the world coordinate system, a translation $T = (U\ V\ W)^T$ applied to a point in 3-D $X = (X\ Y\ Z)^T$ is accomplished through vector addition:

$$X' = T + X = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} + \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (1)$$

A 3-D rotation R about an arbitrary axis through the origin of coordinate system 130 can be described by successive rotations about its three axes:

$$R = R_\phi R_\theta R_\psi \text{ where} \quad (2)$$

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \text{ rotation about the X-axis,} \quad (3a)$$

$$R_\theta = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \text{ rotation about the Y-axis,} \quad (3b)$$

$$R_\psi = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ rotation about the Z-axis.} \quad (3c)$$

A general rigid motion in space consisting of translation and rotation is described by the transformation
$M: X \rightarrow X' = R_\phi R_\theta R_\psi (T + X)$ (4)
Its six degrees of freedom are U, V, W, ϕ, θ, ψ.

This decomposition is not unique because the translation could be as well applied after the rotation. Also, since the multiplication of the rotation matrices is not commutative, a different order of rotations would result in different amounts of rotation for each axis. For a fixed order of application, however, this motion decomposition is unique.

To model the movements of the vehicle, the camera is considered as being stationary and the environment as being moving as one single rigid object relative to the camera. The origin 0 of coordinate system 130 is located in the lens center 126 of the camera.

The given task is to reconstruct the vehicle's or the camera's egomotion from visual information. It is therefore necessary to know the effects of different kings of vehicle or camera motion upon the camera image. Under perspective imaging, a point in space $X = (X\ Y\ Z)^T$ is projected onto a location on the image plane $x = (X\ Y)^T$ such that
$x = f\frac{X}{Z}\ y = f\frac{Y}{Z}$. (5)
where f is the focal length of the camera (see Figure 3).

The effects of pure camera rotation are accounted for. Ignoring the boundary efforts when the camera is rotated around its lens center 126, the acquired image changes but no new views of the environment are obtained. Pure camera rotation merely maps the image into itself. The most intuitive effect results from pure rotation about the Z-axis of the camera-centered coordinate system 130, which is also the optical axis. Any point in the image moves along a circle centered at the image location $x = (0\ 0)$. In practice, however, the amount of rotation ψ of the vehicle about the Z-axis is small. Therefore, vehicle rotation is confined to the X- and Y-axis, where significant amounts of rotation occur.

The vehicle or camera undergoing rotation about the X-axis by an angle $-\phi$ and the Y-axis by an angle $-\theta$ moves each 3-D point X to point X' relative to the camera.

$$X \rightarrow X' = R_\phi \cdot R_\theta \cdot X \quad (6)$$

$$= \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ \sin\phi\sin\theta & \cos\phi & -\sin\phi\cos\phi \\ -\cos\phi\sin\theta & \sin\phi & \cos\phi\cos\theta \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

iently x, the image point of X, moves to x' given by

$$x' = f \frac{X \cos\theta + Z \sin\theta}{-X \cos\phi \sin\theta + Y \sin\phi + Z \cos\phi \cos\theta} \quad (7a)$$

$$y' = f \frac{X \sin\phi \sin\theta + Y \cos\phi - Z \sin\phi \cos\theta}{-X \cos\phi \sin\theta + Y \sin\phi + Z \cos\phi \cos\theta} \quad (7b)$$

the perspective transformation for the original image point x yields.
x and Y = $\frac{1}{f}$ Z y. (8)
rotation mapping r_θ r_ϕ which moves each image point x = (x y) into the corresponding image = (x' y') under camera rotation R_ϕ R_θ (i.e., a particular sequence of "pan" and "tilt" is given by

$$R_\phi R_\theta (X) : X \rightarrow X'$$

$$r_\phi r_\theta (x) : x = (x\ y) \rightarrow x' = (x'\ y')$$

$$x' = f \frac{x \cos\theta + f \sin\theta}{-x \cos\phi \sin\theta + y \sin\phi + f \cos\phi \cos\theta} \quad (9a)$$

$$x \sin\phi \sin\theta + y \cos\phi - f \sin\phi \cos$$

important to notice that this transformation contains no 3-D variables and is therefore a mapping of ge onto itself. This demonstrates that no additional information about the 3-D structure of the scene obtained under pure camera rotation.
interesting property of this mapping should be mentioned at this point, which might not be obvious. an image point on a diagonal passing through the center of the image at 45° by only rotating the does not result in equal amounts of rotation about the X- and the Y-axis. This is again a uence of the successive application of the two rotations R_θ and R_ϕ, since the first rotation about the also changes the orientation of the camera's X-axis in 3-D space. It also explains why the pair of ns in (7) is not symmetric with respect to θ and ϕ.
measuring the amount of camera rotation, the problem to be solved is the following: Given are two locations x_0 and x_1, which are the observations of the same 3-D point at time t_0 and time t_1. The n here is the amount of rotation R_ϕ and R_θ which applied to the camera between time t_0 and time ild move image point x_0 onto x_1 assuming that no camera translation occurred at the same time. If R_θ are applied to the camera separately, the points in the image move along hyperbolic paths. If orizontal rotation were applied to the camera, a given image point x_0 would move on a path ›ed by

$$r_\theta (x_0): y^2 = y^2_0 \frac{f^2 + x^2}{f^2 + x^2_0} \quad (10)$$

ly pure vertical camera rotation would move an image point x_1 along

$$r_\phi (x_1): x^2 = x^2_1 \frac{f^2 + y^2}{f^2 + y^2_1} \quad (11)$$

nce the 3-D rotation of the camera is modeled as being performed in two separate steps (R_θ ed by R_ϕ), the rotation mapping r_ϕ r_θ can also be separated into r_θ followed by r_ϕ. In the first step,

applying pure (horizontal) rotation around the Y-axis r_θ, point x_0 is moved to an intermediate image location x_c. The second step, applying pure (vertical) rotation around the X-axis r_Φ, takes point x_c to the final image location x_1. This can be expressed as

$r_{\Phi\theta} = r_\Phi r_\Phi$, where

$r_\theta: x_0 = (x_0\ y_0) \rightarrow x_c = (x_c\ y_c)$

$r_\Phi: x_c = (x_c\ y_c) \rightarrow x_1 = (x_1\ y_1)$ (12)

Figure 4 reveals a successive application of horizontal and vertical rotation. Image point x_0 is to be moved to location x_1 by pure horizontal and vertical camera rotation. Horizontal rotation (about the Y-axis) is applied first, moving x_0 to x_c, which is the intersection point of the two hyperbolic paths for horizontal and vertical rotation. In a second step, x_c is taken to x_1. Then the two rotation angles θ and Φ are found directly. Further in Figure 4, the image point $x_c = (x_c\ y_c)$ is the intersection point of the hyperbola passing through x_0 resulting from horizontal camera rotation (10) with the hyperbola passing through x_1 resulting from vertical camera rotation (11). Intersecting the two hyperbolae gives the image point x_c, with

$$x_c = f\,x_1 \left[\frac{f^2 + x_0^2 + y_0^2}{(f^2 + x_0^2)(f^2 + y_1^2) - x_1^2\,y_0^2} \right]^{1/2} \tag{13a}$$

$$y_c = f\,y_0 \left[\frac{f^2 + x_1^2 + y_1^2}{(f^2 + x_0^2)(f^2 + y_1^2) - x_1^2\,y_0^2} \right]^{1/2} \tag{13b}$$

The amount of camera rotation necessary to map x_0 onto x_1 by applying R_θ followed by R_Φ is finally obtained as

$$\theta = \tan^{-1}\frac{x_c}{f} - \tan^{-1}\frac{x_0}{f} \tag{14}$$

$$\phi = \tan^{-1}\frac{y_c}{f} - \tan^{-1}\frac{y_1}{f}. \tag{15}$$

When the vehicle or camera undergoes pure translation between time t and time t′, every point on the vehicle is moved by the same 3-D vector $T = (U\ V\ W)^T$. Again, the same effect is achieved by keeping the camera fixed and moving every point X_i in the environment to X_i' by applying -T.

Since every stationary point in the environment undergoes the same translation relative to the camera, the imaginary lines between corresponding points X_iX_i' are parallel in 3-D space.

It is a fundamental result from perspective geometry that the images of parallel lines pass through a single point in the image plane called a "vanishing point." When the camera moves along a straight line, every (stationary) image point seems to expand from this vanishing point or contract towards it when the camera moves backwards. This particular image location is therefore commonly referred to as the focus of expansion (FOE) or the focus of contraction (FOC). Each displacement vector passes through the FOE creating the typical radial expansion pattern shown in Figure 5. Figure 5 reveals the location of the FOE. With pure vehicle translation, points in the environment (A,B) move along 3-D vectors parallel to the vector pointing from lens center 126 to the FOE in camera plane 128 (Figure 3). These vectors form parallel lines in space which have a common vanishing point (the FOE) in the perspective image.

As can be seen in Figure 5, the straight line passing through the lens center of the camera and the FOE is also parallel to the 3-D displacement vectors. Therefore, the 3-D vector OF points in the direction of camera translation in space. Knowing the internal geometry of the camera (i.e., the focal length f), the direction of vehicle translation can be determined by locating the FOE in the image. The actual translation vector T applied to the camera is a multiple of the vector $\vec{OF}$ which supplies only the direction of camera translation but not its magnitude. Therefore,

$T = \lambda\ \vec{OF} = \lambda[x_f\ y_f\ f]^T,\ \lambda \epsilon R.$ (16)

Since most previous work incorporated a velocity-based model of 3-D motion, the focus of expansion

: commonly been interpreted as the direction of instantaneous heading, i.e., the direction of vehicle ıslation during an infinitely short period in time. When images are given as "snapshots" taken at discrete :ances of time, the movements of the vehicle must be modeled accordingly as discrete movements from ; position in space to the next. Therefore, the FOE cannot be interpreted as the momentary direction of ıslation at a certain point in time, but rather as the direction of accumulated vehicle translation over a iod of time.

Figure 6 illustrates the concept of FOE for discrete time steps. The motion of vehicle 132 between two nts in time can be decomposed into a translation followed by a rotation. The image effects of pure ıslation (FOE_a) are observed in image I_0. Figure 6 shows the top view of vehicle 132 traveling along a ved path at two instances in time t_0 and t_1. The position of the vehicle 132 in space is given by the ;ition of a reference point on the vehicle P and the orientation of the vehicle is Ω. Figure 6 also displays adopted scheme of 3-D motion decomposition. First, the translation T is applied which shifts the ıicle's reference point (i.e., lens center 126 of the camera) from position P_0 to position P_1 without ınging the vehicle's orientation Ω. The 3-D translation vector T intersects image plane 128 at FOE_a. In the :ond step the vehicle is rotated by ω to the new orientation Ω_1. Translation T transforms image I_0 into ıge I'_1, which again is transformed into I_1 by rotation ω. The important fact is that FOE_a is observed at transition from image I_0 to image I'_1, which is obtained by derotating image I_1 by $-\omega$. Throughout the ;sent description, this scheme (Figure 6) is used as a model for vehicle or camera motion.

The amount of camera translation can be measured. Figure 7 shows the geometric relationships for the) case. The amount of expansion from the FOE for discrete time steps is illustrated. Figure 7 can be ısidered as a top view of the camera, i.e., a projection onto the X/Z-plane of the camera-centered ɔrdinate system 130. The cross section of the image plane is shown as a straight line. The camera ıves by a vector T in 3-D space, which passes through lens center 126 and the FOE in camera plane 3. The 3-D Z-axis is also the optical axis of the camera. The camera is translating from left to right in the ɜction given by $T = (X_f\ f)^T$.

A stationary 3-D point is observed at two instances of time, which moves in space relative to the ıera from X to X', resulting in two images x and x'.

$$X = \begin{bmatrix} X \\ Z \end{bmatrix} \quad \text{and} \quad X' = \begin{bmatrix} X' \\ Z \end{bmatrix} = \begin{bmatrix} X - \Delta X \\ Z - \Delta Z \end{bmatrix} \tag{17}$$

ing the inverse perspective (8) transformation yields

$= \frac{f}{x} X$ and

$= Z - \Delta Z = \frac{f}{x'} X' = \frac{f}{x} (X - \Delta X).$ (18)

ɔm similar triangles (shaded in Figure 7)

$$\frac{\Delta X}{x_f} = \frac{\Delta Z}{f}, \tag{19}$$

d therefore

$$Z = \Delta Z \frac{x' - x_f}{x' - x} = \Delta Z \left(1 + \frac{x - x_f}{x' - x}\right). \tag{20}$$

Thus, the rate of expansion of image points from the FOE contains direct information about the distance the corresponding 3-D points from the camera. Consequently, if the vehicle is moving along a straight e and the FOE has been located, the 3-D structure of the scene can be determined from the expansion ıttern in the image. However, the distance Z of a 3-D point from the camera can only be obtained up to ɜ scale factor ΔZ, which is the distance that the vehicle advanced along the Z-axis during the elapsed ne.

When the velocity of the vehicle (ΔZ / t) in space is known, the absolute range of any stationary point ın be computed. Alternatively, the velocity of the vehicle can be obtained if the actual range of a point in e scene is known (e.g., from laser range data). In practice, of course, any such technique requires that the

FOE can be located in a small area, and the observed image points exhibit significant expansion away from the FOE. As shown below, imaging noise and camera distortion pose problems in the attempt to assure that both of the above requirements are met.

If a set of stationary 3-D points $\{ (X_i, X'_i) \}$ is observed, then of course the translation in the Z-direction is the same for every point.

$Z_i - Z'_i = Z_j - Z'_j = \Delta Z$ for all i,j. (21)

Therefore, the range of every point is proportional to the observed amount of expansion of its image away from the FOE

$$Z_i = \alpha \frac{x'_i - x_f}{x'_i - x_i}, \qquad (22)$$

which renders the relative 3-D structure of the set of points.

The effects of camera translation T can be formulated as a mapping t of a set of image locations $\{x_i\}$ into another set of image locations $\{x'_i\}$. Unlike in the case of pure camera rotation, this mapping not only depends upon the 3-D translation vector but also upon the actual 3-D location of each individual point observed. Therefore, in general, t is not simply a mapping of the image onto itself. However, one important property of t can be described exclusively in image plane 128, namely that each point must map onto a straight line passing through the original point and one unique location in the image (the FOE). This means that if vehicle 132 is undergoing pure translation, then there must exist an image location x_f such that the mapping t satisfies the condition radial-mapping $t(x_f, I, I')$:

$t = \{ (x_i, x'_i) \in I \times I' \mid x'_i = x_i + \mu_i (x_i - x_f), \mu_i \in R, \mu_i \geq 0 \}$. (23)

When the vehicle is not undergoing pure translation or rotation but combined 3-D motion (i.e., translation and rotation) of the form $R_\Phi R_\theta T$, the effects in the image are described by a transformation d (for displacement) which is a combination of r_Φ, r_θ and t:

$d : I \rightarrow I' = r_\Phi r_\theta t (I)$, (24)

where $I = \{x_i\}$, $I' = \{x'_i\}$ are the two sets of corresponding image points. Figure 8a and 8b show a typical displacement field for a camera undergoing horizontal and vertical rotation as well as translation. The points $x_i \in I$ are marked with small circles 138. Rectangle 134 marks the area of search for the FOE. The derotated displacement is illustrated in Figure 8b with the FOE marked by circle 136.

By decomposing a composite displacement field d into its three components r_Φ, r_θ, and t, the vehicle's rotation and direction of translation in space can be computed from the information available in the image. This problem is addressed below. As discussed in the previous section, the 3-D motion M of the vehicle is modeled by a translation T followed by a rotation R_θ about the Y-axis and a rotation R_Φ about the X-axis:

$M = R_\Phi R_\theta T$. (25)

This results in a mapping d from the original image I_0 at time t_0 into the new image I_1 at time t_1.

$d: I_0 \rightarrow I_1 = r_\Phi r_\theta t I_0 = r_\Phi r_\theta I'_0$. (26)

The intermediate image I'_0 in (26) is the result of the translation component of the vehicle's motion and has the property of being a radial mapping (23). Unlike the two images I_0 and I_1, which are actually given, the image I'_0 is generally not observed, except when the camera rotation is zero. It serves as an intermediate result to be reached during the separation of translational and rotational motion components.

The question at this point is whether there exists more than one combination of rotation mappings r_Φ and r_θ which would satisfy this requirement, i.e., if the solution is unique. It has been pointed out above that the decomposition of 3-D motion into R_Φ, R_θ, R_ψ, and T is unique for a fixed order of application. This does not imply, however, that the effects of 3-D motion upon the perspective image are unique as well.

Related art has shown that seven points in two perspective views suffice to obtain a unique interpretation in terms of rigid body motion and structure, except for a few cases where points are arranged in some very special configuration in space. Further art reports computer experiments which suggest that six points are sufficient in many cases and seven or eight points yield unique interpretations in most cases.

Due to its design and the application, however, the motion of a typical autonomous land vehicle (ALV) in space is quite restricted. The vehicle can only travel upright on a surface and its large wheelbase allows for only relatively small changes in orientation. It is also heavy and thus exhibits considerable inertia. Therefore, the final motion parameters must lie within a certain narrow range and it can be expected that a unique solution can be found even in cases when the number of points is near or above the minimum.

The fact that

$I'_0 = r_\theta^{-1} r_\Phi^{-1} I_1 = t I_0$ (27)

suggests two different strategies for separating the motion components: (1) FOE from rotation – successively apply combinations of inverse rotation mappings

$r_{\theta_1}^{-1} r_{\phi_1}^{-1}, r_{\theta_2}^{-1} r_{\phi_2}^{-1}, r_{\theta_K}^{-1} r_{\phi_K}^{-1}$

to the second image I_1, until the resulting image I' is a radial mapping with respect to the original image I_0. Then locate the FOE x_{I_K} in I_0 and (2) rotation from FOE – successively select FOE-locations (different directions of vehicle translation) $X_{I_1}, X_{I_2}, \ldots X_{I_1}$ in the original image I_0 and then determine the inverse rotation mapping $r_{\theta_1}^{-1} r_{\phi_1}^{-1}$ that yields a radial mapping with respect to the given FOE x_{I_1} in the original image I_0.

Both alternatives were investigated under the assumption of restricted, but realistic vehicle motion, as stated earlier. It turned out that the major problem in the FOE-from-rotation approach is to determine if a mapping of image points is (or is close to being) radial when the location of the FOE is unknown. Of course, in the presence of noise, this problem becomes even more difficult. The second approach was examined after it appeared that any method which extends the given set of displacement vectors backwards to find the FOE is inherently sensitive to image degradations.

Although there have been a number of suggestions for FOE-algorithms in the past, no results of implementations have been demonstrated on real outdoor imagery. One reason for the absence of useful results might be that most researchers have tried to locate the FOE in terms of a single, distinct image location. In practice, however, the noise generated by merely digitizing a perfect translation displacement field may keep the resulting vectors from passing through a single pixel. Even for human observers it seems to be difficult to determine the exact direction of heading (i.e., the location of the FOE on the retina). Average deviation of human judgement from the real direction has been reported to be as large as 10° and up to 20° in the presence of large rotations. It was, therefore, an important premise in this work that the final algorithm should determine an area of potential FOE-locations (called the "fuzzy FOE") instead of a single (but probably incorrect) point.

The FOE may be obtained from rotation. In this method, the image motion is decomposed in two steps. First, the rotational components are estimated and their inverses are applied to the image, thus partially "derotating" the image. If the rotation estimate was accurate, the resulting displacement field after derotation would diverge from a single image location (the FOE). The second step verifies that the displacement field is actually radial and determines the location of the FOE. For this purpose, two problems have to be solved: (1) how to estimate the rotational motion components without knowing the exact location of the FOE, and (2) how to measure the "goodness of derotation" and locate the FOE.

The rotational components can be estimated. Each vector in the displacement field is the sum of vector components caused by camera rotation and camera translation. Since the displacement caused by translation depends on the depth of the corresponding points in 3-D space (equation 18), points located at a large distance from the camera are not significantly affected by camera translation. Therefore, one way of estimating vehicle rotation is to compute θ and ϕ from displacement vectors which are known to belong to points at far distance. Under the assumption that those displacement vectors are only caused by rotation, equations 14 and 15 can be applied to find the two angles. In some situations, distant points are selected easily. For example, points on the horizon are often located at a sufficient distance from the vehicle. Image points close to the axis of translation would be preferred because they expand from the FOE slower than other points at the same depth. However, points at far distances may not always be available or may not be known to exist in the image. In those cases, the following method for estimating the rotational components can be used. The design of the ALV (and most other mobile robots) does not allow rapid changes in the direction of vehicle heading. Therefore, it can be assumed that the motion of the camera between two frames is constrained, such that the FOE can change its location only within a certain range. If the FOE was located in one frame, the FOE in the subsequent frame must lie in a certain image region around the previous FOE location. Figure 9a shows an image plane which illustrates this situation. The FOE of the previous frame was located at the center of square 140 which outlines the region of search for the current FOE, thus the FOE in the given frame must be inside square 140. Three displacement vectore are shown P1 → P1′, P2 → P2′, P3 → P3′). The translational components (P1 → Q1, P2 → Q2, P3 → Q3) of those displacement vectors and the FOE (inside square 140) are not known at this point in time but are marked in Figure 9a.

The main idea of this technique is to determine the possible range of camera rotations which would be consistent with the FOE lying inside marked region 140. Since the camera rotates about two axes, the resulting range of rotations can be described as a region in a 2-D space. Figure 9b shows this rotation space with the two axes θ and ϕ corresponding to the amount of camera rotation around the Y-axis and the X-axis, respectively. The initial rotation estimate is a range of $\pm 10^\circ$ in both directions which is indicated by a square 142 in rotation space of Figure 9b.

In general, the range of possible rotations is described by a closed, convex polygon in rotation space. A particular rotation (θ', ϕ') is possible if its application to every displacement vector (i.e., to its endpoint) yields a new vector which lies on a straight line passing through the maximal FOE-region. The region of possible rotations is successively constrained by applying the following steps for every displacement vector (Figure 10a and 10b). First apply the rotation mapping defined by the vertices of the rotation polygon to the endpoint P' of the displacement vector $P \rightarrow P'$ 146. This yields a set of image points P_i. Second, connect the points P_i to a closed polygon 148 in the image. Polygon 148 is similar to the rotation polygon 144 but distorted by the nonlinear rotation mapping as shown in Figure 10a. Third, intersect polygon 146 in the image with open triangle 150 formed by the starting point P of the displacement vector 146 and defined by two tangents 152 and 154 onto the maximal FOE-region. Rotations that would bring the endpoint of the displacement vector outside triangle 150 are not feasible. The result is a new (possibly empty) polygon 156 in the image plane. Fourth, new polygon 156 from the image plane back into the rotation space of Figure 9b. Fifth, rotation polygon 158 is empty (number of vertices is zero), then stop. No camera rotation is possible that would make all displacement vectors intersect the given FOE-region. Repeat the process using a larger FOE-region.

Figures 11a, 11b and 11c show the changing shape of the rotation polygon during the application of this process to the three displacement vectors in Figure 8.

Since the mapping from rotation space to the image plane is nonlinear (equation 9), the straight lines between vertices in the rotation polygon 160 do not correspond to straight lines in the image. They are, however, approximated as straight lines in order to simplify the intersection with the open triangle. The dotted lines in the image plane show the actual mapping of the rotation polygon onto the image. It can be seen that the deviations from straight lines are small and can be neglected. Figure 11a shows rotation polygon after examining displacement vector $P1 \rightarrow P1'$. Any camera rotation inside the polygon would move the endpoint of the displacement vector ($P1'$) into the open triangle formed by targets 164 and 166 through P1 to the maximal FOE-region given by square 68 in the image plane. The actual mapping of the rotation polygon into the image plane is shown with a dotted outline.

Figure 11b reveals the rotation polygon 170 after examining displacement vectors $P1 \rightarrow P1'$ and $P2 \rightarrow P2'$.

Figure 11(c) shows the final rotation polygon after examining the three displacement vectors $P1 \rightarrow P1'$, $P2 \rightarrow P2'$ and $P3 \rightarrow P3'$. The amount of actual camera rotation ($\theta = -2.0^\circ$, $\phi = 5.0^\circ$) is marked with a small circle (arrow) 172.

Increasing the number of displacement vectors improves the rotation estimate. In practice, the amount of camera rotation can be constrained to a range of below 1° in both directions. Rotation can be estimated more accurately when the displacement vectors are short, i.e., when the amount of camera translation is small. This is in contrast to estimating camera translation which is easier with long displacement vectors.

The situation when the rotation polygon becomes empty requires some additional considerations. As mentioned earlier, in such a case no camera rotation is possible that would make all displacement vectors pass through the given FOE-region. This could indicate one of the two alternatives. First, at least one of the displacement vectors belongs to a moving object. Second, the given FOE-region does not contain the actual location of the FOE, i.e., the region is not feasible. The latter case is of particular importance. If a region can be determined not to contain the FOE, then the FOE must necessarily lie outside this region. Therefore, the above method can not only be used to estimate the amount of camera rotation, but also to search for the location of the FOE. Unfortunately, if the rotation polygon does not become empty, this does not imply that the FOE is actually inside the given region. It only means that all displacement vectors would pass through this region, not that they have a common intersection inside this region. However, if not all vectors pass through a certain region, then this region cannot possibly contain the FOE. The following recursive algorithm searches a given region for the FOE by splitting it into smaller pieces (divide-and-conquer):

```
ASIBLE (region, min-size, disp-vectors):

E (region) < min-size then return (region)


FEASIBLE (region, disp-vectors) then

return (union)

    MIN-FEASIBLE (sub-region-1, min-size,

        disp-vectors),

    MIN-FEASIBLE (sub-region-2, min-size,

        disp-vectors),

    ....

    MIN-FEASIBLE (sub-region-n, min-size,

        disp-vectors)))

eturn (nil) {region does not contain the FOE}
```

smallest feasible FOE-region by systematically discarding sub regions from
:ase that the shape of the original region is a square, subregions can be
into four subsquares of equal size. The simple version shown here performs
ɜ smallest subregion (limited by the parameter "min-size"), which is neither
fficient approach. The algorithm can be significantly improved by applying a
example, by trying to discard subregions around the perimeter first before
ɔn. Two major problems were encountered with the latter method. First, the
ɔensive since the process of computing feasible rotations must be repeated
ı small region is more likely to be discarded than a larger one. However,
comes too small, errors induced by noise, distortion, or point-tracking may
ɔm passing though a region which actually contains the FOE.
ıot employed in the further treatment, it suggests an interesting alternative
n traditional FOE-algorithms. Its main attractiveness is that it is inherently
most other techniques which search for a single FOE-location. For the
unt of rotation, the method using points at far distance mentioned earlier is
other alternatives for locating the FOE once the rotation components have
n the following.
ally derotated image may be attempted. After applying a particular derotation
field, the question is how close the new displacement field is to a radial
erge from one image location. If the displacement field is really radial, then
ted and only the components due to camera translation remain. Two different
ɔperty are noted. One method uses the variance of intersection at imaginary
.e second method computes the linear correlation coefficient to measure how
ɗ is. The variance of intersection in related art suggests to estimate the
ıt field by computing the variance of intersections of one displacement vector
ntersections lie in a small neighborhood, then the variance is small, which
t field is almost radial. The problem can be simplified by using an imaginary
ɜad, whose orientation is not affected by different camera rotations. Figure 12
$P_1 \rightarrow P_1' \ldots P_5 \rightarrow P_5'$ intersecting a vertical line at x at $y_1 \ldots y_5$. Moving the vertical
ng the points of intersection closer together and will thus result in a smaller
oint of intersection of a displacement vector $P_i \rightarrow P_i'$ with a vertical line at x is

given by

$$\bar{y}_i = \frac{x_i y'_i - y_i x'_i}{x_i - x'_i} \qquad (28)$$

The variance of intersection of all displacement vectors with the vertical line at position x is

$$\sigma^2(x) = \frac{1}{N}\left[\sum_{i,\, x_i \neq x'_i} \bar{y}^2_i - \frac{1}{N}\left[\sum_{i,\, x_i \neq x'_i} \bar{y}_i\right]^2\right]. \qquad (29)$$

To find the vertical cross section with minimum intersection variance, the first derivative of (29) with respect to x is set to zero. The location x_0 of minimum intersection variance is then obtained. Similarly, the position of a horizontal cross section with minimal intersection variance can be obtained.

The square root of the variance of intersection (standard deviation) at a vertical line was evaluated on the synthetic displacement field shown in Figure 13. The actual FOE is located in the center of the image. Square 174 around the center (±100 pixels in both directions) marks the region over which the error functions are evaluated.

Figures 14a, b, c and d show the distribution of the intersection standard deviation for increasing residual rotations in vertical direction in the absence of noise. The horizontal rotation is 0° in all cases represented by these Figures. Locations of displacement vectors are represented by real numbers (not rounded to integer values). In Figure 14a, no residual rotation exists, i.e., the displacement field is perfectly radial. The value of the horizontal position of the cross section varies ±100 pixels around the actual FOE. The standard deviation is zero for $x = x_f$ (the x-coordinate of the FOE) and increases linearly on both sides of the FOE. In Figures 14b-d, the residual vertical rotation is increased from 0.2° to 1.0°. The bold vertical bar marks the horizontal position of minimum standard deviation, the thin bar marks the location of the FOE (X_f). It can be seen that the amount of minimum standard deviation rises with increasing disturbance by rotation, but that the location of minimum standard deviation does not necessarily move away from the FOE.

Figures 15-17 show the same function under the influence of noise. In Figure 15a-d, noise was applied except that by merely rounding the locations of displacement vectors to their nearest integer values. These Figures show standard deviation of intersection (square root) at a vertical cross section at position x for different amounts of vertical rotation with no horizontal rotation. Uniform noise of ±1 pixels was added to the image locations in Figures 16a-d. In Figures 17a-d, uniform noise of ±2 pixels was applied to the image locations. It can be seen that the effects of noise are similar to the effects caused by residual rotation components. The purpose of this error function is to determine where the FOE is located, and how "radial" the current displacement field is.

If the displacement field is already perfectly derotated, then the location of minimum intersection standard deviation is the location of the FOE. Ideally, all vectors pass through the FOE, such that a cross section through the FOE yields zero standard deviation. The question is how well the FOE can be located in an image which is not perfectly derotated. Figure 18a-d plot the location of minimum intersection standard deviation under varying horizontal rotation. The vertical rotation is kept fixed for each plot. Horizontal camera rotations from -1° to +1° are shown on the abscissa (rot). The ordinate (x0) gives the location of minimum standard deviation in the range of ±100 pixels around the FOE (marked xf). The location of minimum standard deviation depends strongly on the amount of horizontal rotation.

A problem is that the location of minimum standard deviation is not necessarily closer to the FOE when the amount of rotation is less. The function is only well behaved in a narrow range around zero rotation, which means that the estimate of the camera rotation must be very accurate to successfully locate the FOE. The second purpose of this error function is to measure how "radial" the displacement field is after partial derotation. This should be possible by computing the amount of minimum intersection standard deviation. Intuitively, a smaller amount of minimum intersection standard deviation should indicate that the displacement field is less disturbed by rotation. Figures 19a-d and 20a-d show that this is generally true by showing the amount of minimum intersection standard deviation under varying horizontal rotation. For the noise-free case in Figure 19a, the amount of minimum intersection standard deviation becomes zero in the absence of horizontal and vertical rotations, indicating that the derotation is perfect. Unfortunately, the function is not

well behaved even in this relatively small range of rotations (±1.0°). The curve exhibits some sharp local minima where an algorithm searching for an optimal derotation would get trapped easily. Figures 19a-d show the same function in the presence of ±2 pixels of uniform noise added to image locations.

The second method, utilizing linear correlation, of measuring how close a displacement field is to a radial pattern again uses the points y_{11}-y_{15} and y_{21}-y_{25} of intersection at vertical (or horizontal) lines x_1 and x_2 as illustrated in Figure 21. The displacement vectors $P_1 \rightarrow P'_1$ through $P_5 \rightarrow P'_5$ are intersected by two vertical lines X_1 and X_2, both of which lie on the same side of the FOE that is within area 176. Since the location of the FOE is not known, the two lines X_1 and X_2 are simply located at a sufficient distance from any possible FOE-location. This results in two sets of intersection points $\{ (x_1, y_{1i}) \}$ and $\{ (x_2, y_{2i}) \}$. If all displacement vectors emanate from one single image location, then the distances between corresponding intersection points in the two sets must be proportional, i.e.,

$$\frac{y_{1i} - y_{1j}}{y_{2i} - y_{2j}} = \frac{y_{1j} - y_{1k}}{y_{2j} - y_{2k}} \quad \text{for all } i,j,k. \qquad (30)$$

Therefore, a linear relationship exists between the vertical coordinates of intersection points on these two lines. The "goodness" of this linear relationship is easily measured by computing the correlation coefficient for the y-coordinates of the two sets of points. The resulting coefficient is a real number in the range from -1.0 to +1.0. If both vertical lines are on the same side of the FOE, then the optimal value is +1.0. Otherwise, if the FOE lies between the two lines, the optimal coefficient is -1.0. The horizontal position of the two vertical lines is of no importance, as long as one of these conditions is satisfied. For example, the left and right border lines of the image can be used.

Figures 22a-d and 23a-d show plots for the correlation coefficient for intersection of displacement vectors at two vertical lines under varying horizontal rotations, under the same conditions as in Figures 19a-d and 20a-d. No noise was applied for Figure 22a-d. In Figures 23a-d, a uniform noise of ±2 pixels was added to the image locations. The optimal coefficient is + 1.0 (horizontal axis) in Figures 22a-d and 23a-d. The shapes of the curves of Figures 22a-d and 23a-d are similar, respectively, to Figures 19a-d and 20a-d for the minimum standard deviations shown above earlier, with peaks at the same locations. It is apparent, however, that each curve has several locations where the coefficient is close to the optimum value (+1.0), i.e., no distinct global optimum exists which is not only the case in the presence of noise (Figures 23a-d). This fact makes the method of maximizing the correlation coefficient useless for computing the FOE.

The main problem encountered in computing the FOE from rotation, just described above, is that none of the functions examined was well behaved, making the search for an optimal derotation and the location of the FOE difficult. Disturbances induced by noise and residual rotation components are amplified by extending short displacement to straight lines and computing their intersections. The method, rotation from FOE, described below avoids this problem by guessing an FOE-location first and estimating the optimal derotation for this particular FOE in the second step.

Given the two images I_0 and I_1 of corresponding points, the main algorithmic steps of this approach are: (1) Guess an FOE-location $x_f^{(i)}$ in image I_0 (for the current iteration i); (2) Determine the derotation mapping r_θ^{-1}, r_ϕ^{-1} which would transform image I_1 into an image I'_1 such that the mapping $(x_f^{(i)}, I_0, I'_1)$ deviates from a radial mapping (equation 23) with minimum error $E^{(i)}$; and (3) Repeat steps (1) and (2) until an FOE-location $x_f^{(k)}$ with the lowest minimum error $E^{(k)}$ is found.

An initial guess for the FOE-location is obtained from knowledge about the orientation of the camera with respect to the vehicle. For subsequent pairs of frames, the FOE-location computed from the previous pair can be used as a starting point.

Once a particular x_f has been selected, the problem is to compute the rotation mappings r_θ^{-1} and r_ϕ^{-1} which, when applied to the image I_1, will result in an optimal radial mapping with respect to I_0 and x_f.

To measure how close a given mapping is to a radial mapping, the perpendicular distances between points in the second image (x'_i) and the "ideal" displacement vectors is measured. The "ideal" displacement vectors lie on straight lines passing through the the FOE x_f and the points in the first image x_i (see Figure 24) which illustrates measuring the perpendicular distance d_i between lines from x_f through points x_i in the second image. The sum of the squared perpendicular distances d_i, is the final error measure. For each set of corresponding image points ($x_i \in I$, $x'_i \in I'$), the error measure is defined as

$$E(x_f) = \sum_i E_i = \sum_i d^2_i = \sum_i \left[\frac{1}{|\vec{x_f x_i}|} \; \vec{x_f x_i} \times \vec{x_f x'_i} \right]^2 . \qquad (31)$$

In the following, it is assumed that the amount of residual image rotation in horizontal and vertical direction is moderately small (less than 4°). In most practical cases, this condition is satisfied, provided that the time interval between frames is sufficiently small. However, should the amount of vehicle rotation be very large for some reason, a coarse estimate of the actual rotation can be found (as described above) and applied to the image before the FOE computation. With small amounts of rotation, the actual rotation mapping, where points move on horizontal and vertical hyperbolic paths, can be approximated by a horizontal and vertical shift with constant length over the entire image. Under this condition, the inverse rotation mapping r_Φ^{-1}, r_ϑ^{-1} can be approximated by a adding a constant vector $s = (s_x\ s_y)$ which is independent of the image location:

$I' = r_\vartheta^{-1} r_\Phi^{-1} I = s + I$. (32)

Given two images I and I' the error measure (equation 31) becomes

$$E(x_f, s) = \sum_i \left[\frac{1}{|\vec{x_f x_i}'|} \left[\vec{x_f x_i} \times (\vec{x_f x'_i} + s) \right]^2 \right] \qquad (33)$$

where $x_i \in I$ and $x'_i \in I$. For a given FOE-location x_f, the problem is to minimize E with respect to the two unknowns s_x and s_y. To reduce this problem to a one-dimensional search, one point x_g, called the "guiding point, is selected in image I which is forced to maintain zero error (see Figure 25) wherein one vector x_g is selected from the set of displacement vectors to determine the optimum 2-D shift to be applied to points x'_i, given a FOE-location x_f.

First x'_g is forced onto the line $x_f x_g$ and then the entire image $I' = \{x'_1, x'_2, \ldots\}$ is translated in the direction of this line until the error value reaches a minimum. Therefore, the corresponding point x'_g must lie on a straight line passing through x_f and x_g. Any shift s applied to the image I' must keep x'_g on this straight line, so

$x'_g + s = x_f + \lambda (x_g - x_f)$ for all s, (34a)

for all s, and thus,

$s = x_f - x'_g + \lambda (x_g - x_f) (\lambda \in R)$. (34b)

For $\lambda = 1$, $s = x_g - x'_g$, which is the vector $x'_g \rightarrow x_g$. This means that the image I' is shifted such that x_g and x'_g overlap. This leaves λ as the only free variable and the error function (equation 33) is obtained as

$$E(\lambda) = \sum_i \left[\lambda A_i + B_i - C_i \right]^2 \qquad (35)$$

$$l_{if} = \sqrt{(x_i - x_f)^2 + (y_i - y_f)^2}$$

$$A_i = \frac{1}{l_{if}} (y_i - y_f)(x_g - x_f) - (x_i - x_f)(y_g - y_f)$$

$$B_i = \frac{1}{l_{if}} (y_i - y_f)(x'_i - x'_g)$$

$$C_i = \frac{1}{l_{if}} (x_i - x_f)(y'_i - y'_g) .$$

Differentiating equation 35 with respect to λ and forcing the resulting equation to zero yields the parameter for the optimal shift s_{opt} as

$$\lambda_{opt} = \frac{\sum A_i C_i - \sum A_i B}{\sum A^2_i} \qquad (36)$$

The optimal shift s_{opt} and the resulting minimum error $\Sigma(\lambda_{opt})$ for the given FOE-location x_f is obtained by inserting λ_{opt} into equations (34b) and (35) respectively, giving

$E_{min}(x_f) = \lambda^2_{opt} \Sigma A^2_i + 2\lambda_{opt} [\Sigma A_iB_i - \Sigma A_iC_i] - 2 \Sigma B_iC_i + \Sigma B^2_i + \Sigma C^2_i$ (37)

The normalized error E_n shown in the following results (shown in Figures 27-32) is defined as

$E_n(x_f) = \sqrt{N} E_{min}(x_f)$ (38)

where N is the number of displacement vectors used for computing the FOE.

Since in a displacement field caused by pure camera translation all vectors must point away from the FOE, this restriction must hold for any candidate FOE-location (as illustrated in Figure 25). If after applying $s_{opt}(x_f)$ to the second image I', the resulting displacement field contains vectors pointing towards the hypothesized x_f, then this FOE-location is prohibited and can be discarded from further consideration such is the case at point x_1 in Figure 26. Figure 26 shows a field of 5 displacement vectors. The optimal shift s_{opt} for the given x_f is shown as a vector in the lower right-hand corner. When s_{opt} is applied to point x'_1, the resulting displacement vector (shown fat) does not point away from the FOE. Since its projection onto the line x_fx_1 points towards the FOE, it is certainly not consistent with a radial expansion pattern.

The final algorithm for determining the direction of heading as well as horizontal and vertical camera rotations is the "find-FOE algorithm" which consists the steps: (1) Guess an initial FOE x_f^0, for example the FOE-location obtained from the previous pair of frames; (2) Starting from x_f^0, search for a location x_f^{opt} where $E_{min}(x_f^{opt})$ is a minimum. A technique of steepest descent is used, where the search proceeds in the direction of least error; and (3) Determine a region around x_f^{opt} in which the error is below some threshold. The search for this FOE-area is conducted at FOE-locations lying on a grid of fixed width. In the examples shown, the grid spacing is 10 pixels on both x- and y-directions.

The error function $E(x_f)$ is computed in time proportional to the number of displacement vectors N. The final size of the FOE-area depends on the local shape of the error function and can be constrained not to exceed a certain maximum M. Therefore, the time complexity is O(MN).

The first set of experiments was conducted on synthetic imagery to investigate the behavior of the error measure under various conditions, namely, the average length of the displacement vectors (longer displacement vectors lead to a more accurate estimate of the FOE), the amount of residual rotation components in the image, and the amount of noise applied to the location of image points. Figures 27a-d shows the distribution of the normalized error $E_n(x_f)$ for a sparse and relatively short displacement field containing 7 vectors. Residual rotation components of $\pm 2^\circ$ in horizontal and vertical direction are present in Figures 27b-d to visualize their effects upon the image. This displacement field was used with different average vector lengths (indicated as length-factor) for the other experiments on synthetic data. The displacement vector through the guiding point is marked with a heavy line. The choice of this point is not critical, but it should be located at a considerable distance from the FOE to reduce the effects of noise upon the direction of the vector $x_f\vec{x}_g$. Figures 27a-d, which show displacement field and minimum error at selected FOE-locations, the error function is sampled in a grid with a width of 10 pixels over an area of 200 by 200 pixels around the actual FOE, which is marked by small square 178. At each grid point, the amount of normalized error is (equation 41) indicated by the size of the circle 180. Heavy circles 180 indicate error values which are above a certain threshold. Those FOE-locations that would result in displacement vectors which point towards the FOE (as described above) are marked as prohibited (+). It can be seen that the shape of the 2-D error function changes smoothly with different residual rotations over a wide area and exhibits its minimum close to the actual location of the FOE. Figure 27a represents no residual rotation, Figure 27b represents 2.0° of horizontal camera rotation (to the left), Figure 27c represents 2.0° of vertical rotation (upwards), and Figure 27d represents -2.0° vertical rotation (downwards).

Figures 28 to 33 show the effects of various conditions upon the behavior of this error function in the same 200 by 200 pixel square around the actual FOE as in Figures 27a-d.

Figures 28a-e show how the shape of the error function depends upon the average length (with length factors varying from 1 to 15) of the displacement vectors in the absence of any residual rotation or noise (except digitization noise). The minimum of the error function becomes more distinct with increasing amounts of displacement. Figures 29a-e show the effect of increasing residual rotation in horizontal direction upon the shape of the error function for relatively short vectors (length factor of 2.0) in absence of noise.

Figures 30a-e show the effect of residual rotation in vertical direction upon the shape of the error function for short vectors (length factor of 2.0) in absence of noise. Here, it is important to notice that the displacement field used is extremely nonsymmetric along the Y-axis of the image plane. This is motivated by the fact that in real ALV images, long displacement vectors are most likely to be found from points on the ground, which are located in the lower portion of the image. Therefore, positive and negative vertical rotations have been applied in Figures 30a-e.

In Figures 31a-j, residual rotations in both horizontal and vertical direction, respectively, are present, for short vectors with a length factor of 2.0. In Figures 31a-e, the error function is quite robust against rotational components in the image. Figures 31f-j show the amounts of optimal linear shift s_{opt} under the same conditions. The result in Figure 31e shows the effect of large combined rotation of 4.0° 4.0° in both directions. Here, the minimum of the error function is considerably off the actual location of the FOE because of the error induced by using a linear shift to approximate the nonlinear derotation mapping. In such a case, it would be necessary to actually derotate the displacement field by the amount of rotation equivalent to s_{opt} found at the minimum of this error function and repeat the process with the derotated displacement.

The effects of various amounts of uniform noise applied to image point coordinates for a constant average vector length of 5.0, are shown in Figures 32a-e. For this purpose, a random amount (with uniform distribution) of displacement was added to the original (continuous) image location and then rounded to integer pixel coordinates. Random displacement was applied in ranges from ±0.5 to ±4.0 pixels in both horizontal and vertical direction. The shape of the error function becomes flat around the local minimum of the FOE with increasing levels of noise. The displacement field contains only 7 vectors. What is observed here is that the absolute minimum error increases with the amount of noise. Figures 32a-e can thus serve as an indicator for the amount of noise present in the image and the reliability of the final result.

The length of the displacement vectors is an important factor. The shorter the displacement vectors are, the more difficult it is to locate the FOE correctly in the presence of noise. Figures 33a and 33b show the error functions for two displacement fields with different average vector lengths (length factor 2.0 and 5.0, respectively). For the shorter displacement field (length-factor 2.0) in Figure 33, the shape of the error function changes dramatically under the same amount of noise (compare Figure 31a). A search for the minimum error (i.e., local minimum) inevitably converge towards an area 182 indicated by the small arrow, far off the actual FOE. For the image with length-factor 5.0 (Figure 33b), the minimum of the error function coincides with the actual location of the FOE 178. The different result for the same constellation of points in the Figure 32d is caused by the different random numbers (noise) obtained in each experiment. This experiment confirms that a sufficient amount of displacement between consecutive frames is essential for reliably determining the FOE and thus, the direction of vehicle translation.

The performance of this FOE algorithm is shown below on a sequence of real images taken from a moving ALV. Also, it is shown how the absolute velocity of the vehicle can be estimated after the location of the FOE has been determined. The essential measure used for this calculation is the absolute height of the camera above the ground which is constant and known. Given the absolute velocity of the vehicle, the absolute distance from the camera of 3-D points in the scene can be estimated using equation 20.

Next, velocity over ground may be computed. After the FOE has been computed following the steps outlined above, the direction of vehicle translation and the amount of rotation are known. From the derotated displacement field and the location of the FOE, the 3-D layout of the scene can be obtained up to a common scale factor (equation 20). As pointed out above, this scale factor and, consequently, the velocity of the vehicle can be determined if the 3-D position of one point in space is known. Furthermore, it is easy to show that it is sufficient to know only one coordinate value of a point in space to reconstruct its position in space from its location in the image.

Since the ALV travels on a fairly flat surface, the road can be approximated as a plane which lies parallel to the vehicle's direction of translation (see Figure 34). This approximation holds at least for a good part of the road in the field of view of the camera 184. Figure 34 shows a side view of camera 184 traveling parallel to flat surface 186. Camera 184 advances in direction Z, such that a 3-D point on ground surface 186 moves relative to camera 184 from Z_0 to Z_1. Depression angle Φ can be determined from the location of FOE 188 in image 128. Height of camera 184 above ground surface 186 is given.

Since the absolute height of camera 184 above the ground 186 is constant and known, it is possible to estimate the positions of points 192 and 194 on road surface 186 with respect to the vehicle 132 (of Figure 6) in absolute terms. From the changing distances between points 192 and 194 and camera 184, the actual advancement and speed can be determined.

First, a coordinate system 130 is introduced which has its origin 0 in the lens center 126 of the camera 184. The Z-axis of coordinate system 130 passes through FOE 188 in the image plane 128 and aims in the

direction of translation. The original camera-centered coordinate system (X Y Z) 130 is transformed into the new frame (X' Y' Z') merely by applying horizontal and vertical rotation until the Z-axis lines-up with FOE 188. The horizontal and vertical orientation in terms of "pan" and "tilt" are obtained by "rotating" FOE 188 (x_f y_f) into the center of image 128 (0 0) using equations 14 and 15 in the following:

$$\theta_f = -\tan^{-1} \frac{x_f}{f} \tag{39}$$

$$\phi_f = -\tan^{-1} \left[y_f \frac{f^2}{(f^2 + x_f^2) f^2 - x_f^2 y_f^2} \right]. \tag{40}$$

The two angles θ_f and ϕ_f represent the orientation of camera 184 in 3-D with respect to the new coordinate system (X' Y' Z'). This allows determination of the 3-D orientation of the projecting rays passing through image points y_0 and y_1 by use of the inverse perspective transformation. A 3-D point X in the environment whose image x = (x y) is given, lies on a straight line in space defined by

$$X = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \kappa \begin{bmatrix} \cos\theta_f & \sin\theta_f \sin\phi_f & -\sin\theta_f \cos\phi_f \\ 0 & \cos\phi_f & \sin\phi_f \\ \sin\theta_f & -\cos\theta_f \sin\phi_f & \cos\theta_f \cos\phi_f \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} \tag{41}$$

For points 192 and 194 on road surface 186 of Figure 34, the Y-coordinate is -h which is the height of camera 184 above ground 186. Therefore, the value of κ_s for a point on the road surface (x_s y_s) can be estimated as

$$\kappa_s = \frac{-h}{y_s \cos\theta_f + f \sin\theta_f} \tag{42}$$

and its 3-D distance is found by inserting κ_s into equation 41 as

$$Z_s = -h \frac{x_s \sin\theta_f - y_s \cos\theta_f \sin\phi_f - f \cos\theta_f \cos\phi_f}{y_s \cos\phi_s + f \sin\phi_s} . \tag{43}$$

If a point on the ground is observed at two instances of time, x_s at time t and x'_s at t', the resulting distances from the vehicle Z_s at t and Z'_s at t' yield the amount of advancement ΔZ_s (t, t') and estimated velocity V_s (t, t') in this period as

$\Delta Z_s (t,t') = Z_s - Z'_s$ (44)

$$V_s (t, t') = \frac{Z_s - Z'_s}{t' - t} . \tag{45}$$

Image noise and tracking errors have a large impact upon the quality of the final velocity estimate. Therefore, the longest available displacement vectors are generally selected for this measurement, i.e., those vectors which are relatively close to the vehicle. Also, in violation of the initial assumption, the ground surface is never perfectly flat. In order to partially compensate these errors and to make the velocity estimate more reliable, the results of the measurements on individual vectors are combined. The length of each displacement vector $|x_i - x'_i|$ in the image is used as the weight for its contribution to the final result. Given a set of suitable displacement vectors $S = \{x_i - x'_i\}$, the estimate of the distance traveled by the vehicle is taken as the weighed average of the measurements ΔZ_i on individual vectors

$$\bar{\Delta Z}\ (t,\ t') = \frac{\sum (|x_i - x'_i|\ \Delta Z_i)}{\sum |x_i - x'_i|} \qquad (46)$$

and the final estimate for the vehicle velocity is

$$\bar{V}\ (t,\ t') = \frac{\bar{\Delta Z}}{t' - t}\ . \qquad (47)$$

This computation was applied to a sequence of real images which is described below.

In the following, results of the FOE-algorithm and computation of the vehicle's velocity over ground are shown on a real image sequence taken from the moving ALV. The original sequence wae provided on standard video tape with a frame-rate of 30 per second. Out of this original sequence, images were taken in 0.5 second intervals, i.e., at a frame rate of 2 per second in order to reduce the amount of storage and computation. The images were digitized to a spatial resolution of 512 x 512, using only the Y-component (luminance) of the original color signal.

Figures 35a-i show the edge images of frames 182-190 of an actual test, with points 1-57 being tracked and labeled with ascending numbers. Figures 35a-i show the original image sequence taken from the moving ALV after edge detection and point detection. Selected points 1-57 are located at the lower-left corners of their marks. Figures 35j-p (frames 191-197), which include additional points 58-78, show the original image sequence taken after edge detection and point selection. An adaptive windowing technique was developed as an extension of relaxation labeling disparity analysis for the selection and matching of tracked points. The actual image location of each point is the lower left corner of the corresponding mark. The resulting data structure consists of a list of point observations for each image (time), e.g.,

time t_0: $((p_0\ t_0\ x_0\ y_0)\ (p_2\ t_0\ x_2\ y_2)\ (p_3\ t_0\ x_3\ y_3)\ \ldots)$

time t_1: $((p_1\ t_1\ x_1\ y_1)\ (p_2\ t_1\ x_2\ y_2)\ (p_3\ t_1\ x_3\ y_3)\ \ldots)$

Points are given a unique label when they are encountered for the first time. After the tracking of a point has started, its label remains unchanged until this point is no longer tracked. When no correspondence is found in the subsequent frame for a point being tracked, either because of occlusion, or the feature left the field of view, or because it could not be identified, tracking of this point is discontinued. Should the same point reappear again, it is treated as a new item and given a new label. Approximately 25 points per image have been selected in the sequence shown in Figures 35a-i. In the search for the focus of expansion, the optimal FOE-location from the previous pair of frames is taken as the initial guess. For the very first pair of frames (when no previous result is available), the location of the FOE is guessed from the known camera setup relative to the vehicle. The points which are tracked on the two cars (24 and 33) are assumed to be known as moving and are not used as reference points to compute the FOE, vehicle rotation, and velocity. This information is eventually supplied by the reasoning processes in conjunction with the qualitative scene model (Figure 2).

Figures 36a-p illustrate the displacement vectors and estimates of vehicle motion for the image sequence shown in Figures 35a-p. Shaded area 190 marks the possible FOE locations and circle 178 inside of area 190 is the FOE with the lowest error value. The FOE-ratio measures the flatness of the error function inside area 190.

Also, Figures 36a-p show the results of computing the vehicle's motion for the same sequence as in the previous Figure. Each frame t displays the motion estimates for the period between t and the previous frame t - 1. Therefore, no estimate is available at the first frame (182). Starting from the given initial guess, the FOE-algorithm first searches for the image location, which is not prohibited and where the error function (equation 35) has a minimum.

The optimal horizontal and vertical shift resulting at this FOE-location is used to estimate the vehicle's rotations around the X- and Y-axis. This point, which is the initial guess for the subsequent frame, is marked as a small circle inside the shaded area. The equivalent rotation components are shown graphically on a $\pm 1^\circ$ scale. They are relatively small throughout the sequence such that it was never necessary to apply intermediate derotation and iteration of the FOE-search. Along with the original displacement vectors (solid lines), the vectors obtained after derotation are shown with dashed lines.

After the location with minimum error has been found, it is used as the seed for growing a region of potential FOE-locations. The growth of the region is limited by two restrictions: (1) The ratio of maximum to minimum error inside the region is limited, i.e.,

$E_n^i/E_n^{min} = \rho^i \leq \rho^{lim}$

(see equation 40 for the definition of the error function E_n). No FOE-location for which the error ratio ρ^i exceeds the limit ρ^{lim} is joined to the region. Thus the final size of the region depends on the shape of the error function. In this example, the ratio ρ^{lim} was set at 4.0. Similarly, no prohibited locations (Figure 26) are considered; (2) The maximum size of the region M is given by the given FOE-region region regardless of their error values. The resulting error ratio $\rho^{max} = \max(\rho^i)$ for the points inside the region indicates the shape of the error function for this area. A low value for the ratio ρ^{max} indicates a flat error function. The value for ρ^{max} is shown as FOE-RATIO in every image.

For the computation of absolute vehicle velocity, only a few prominent displacement vectors were selected in each frame pair. The criteria were that the vectors be located below the FOE and that their length be more than 20 pixels. The endpoints of the selected (derotated) vectors are marked with dark dots. The parameter used for the computation of absolute advancement is the height of the camera above the ground, which is 3.3 meters (11 feet).

Figure 37 illustrates hardware embodiment 200 of the present invention. Camera 202 may be a Hitachi television camera having a 48° vertical field of view and a 50° horizontal field of view. Camera 202 may be set at a depression angle of 16.3° below the horion. Image sequences 204 acquired by camera 202 are transmitted to tracking means 206 comprising an I²S processor 208 and a VAX 11/750 computer 210. Tracking means 206 tracks tokens between frames of image sequences 204. The output of means 206 goes to means 212 for matching of tokens and corresponding images. For the two different computers 210 and 218, a VAX-Symbolics bidirectional network protocol means 214 is connected between means 212 and means 216 which includes Symbolics 3670 computer 218, though it is possible to use one computer thereby eliminating means 214. Computer 218 provides processing for obtaining fuzzy focus of expansion and rotations/translation estimates of motion. The language environment used with computer 218 is Common LISP.

The methods and processes in the embodiment of the present invention are implemented with the ensuing programming.

Claims

1. A method for determining self-motion of an imaging system in an environment, **characterized by:**
computing a focus of expansion (FOE) location from two-dimensional displacement vectors of successive two-dimensional images;
computing a fuzzy FOE region that is a qualitative indication of the FOE location, wherein the fuzzy FOE region is an area of possible FOE locations for each image;
determining an approximate direction of heading and amount of rotation of said imaging system, relative to its own reference coordinate system, from the fuzzy FOE; and
removing the effects of rotation of said imaging system from the displacement vectors.

2. Method according to claim 1 further **characterized by:**
acquiring the successive two-dimensional images of the environment of said imaging system;
selecting features in the images; and determining the two-dimensional displacement vectors for the features of each pair of the successive images.

3. An imaging apparatus capable of determining its own self-motion from its imaging, **characterized by:**
imaging means (202,204) for detecting a sequence of images in a field of view;
token tracking means (206), connected to said imaging means, for determining two-dimensional displacement vectors of selected tokens for each consecutive pair of images, and wherein said token tracking means tracks stationary parts of a visual environment in the images, using two-dimensional, references such as corner points, contour segments, region boundaries as references, from image to image, wherein the two-dimensional displacement vectors are the result of camera motion;
seeker means (208,210), connected to said token tracking means, for selecting candidate locations for focus of expansion under forward translation of said imaging means and for focus of contraction under backward translation of said imaging device, for forming a connected image region of the candidate locations and a range of corresponding rotations, and for outputting focus of expansion or contraction and image rotation estimates; and
optimal rotation means (218), connected to said token tracking means and to said seeker means, for determining optimal three-dimensional rotation angles plus an error value for a selected candidate location for focus of expansion or contraction.

4. Apparatus according to claim 3. **characterized in that:**
said seeker means (208.210) receives signals conveying the two-dimensional displacement vectors from said token tracking means: and
said optimal rotation means (218) receives signals conveying the two-dimensional displacement vectors from said token tracking means and signals conveying a single focus of expansion or contraction location from said seeker means. and sends the determined optimal three-dimensional rotation angles (pan and tilt) plus the error value for the single focus of expansion or contraction location. to said seeker means.

5. Apparatus according to claim 4. **characterized in that** said optimal rotation means (218) determines the optimal three-dimensional rotation angles (pan and tilt) for the single focus of expansion (direction of heading) or contraction location. by simulating the effects of reverse imaging means rotations upon a given set of displacement vectors from said token tracking means. and thus virtually rotating said sensing device until a modified displacement field approaches closest to a radial expansion pattern relative to the selected candidate location for focus of expansion or contraction. the optimal three-dimensional (pan and tilt) rotation angles indicating needed reverse rotation of said imaging means and the error value. wherein said error value is a deviation from radial displacement field.

6. Apparatus according to claim 5. **characterized in that** said optimal rotation means determines velocity of said imaging device.

7. An imaging system capable of determining its self-motion from its imaging. **characterized by:**
derotation means (122) for derotating two-dimensional displacement vectors from consecutive pairs of two-dimensional images to remove rotational effects of said imaging system;
first computing means (124). connected to said derotation means. for computing a fuzzy focus of expansion (FOE) from the two-dimensional displacement vectors. wherein the fuzzy FOE is a two-dimensional region of possible focus-of-expansion locations on a two-dimensional image; and
second computing means (118). connected to said derotation means. for computing self-motion parameters of said imaging system. from the fuzzy FOE.

8. System according to claim 7 further **characterized by:**
sensing means (202,112) acquiring the two-dimensional images;
feature detection means (114). connected to said sensing means. for detecting. extracting and tracking features in the two dimensional images: and
third computing means (118). connected to said feature detection means. for computing the two dimensional displacement vectors from the features.

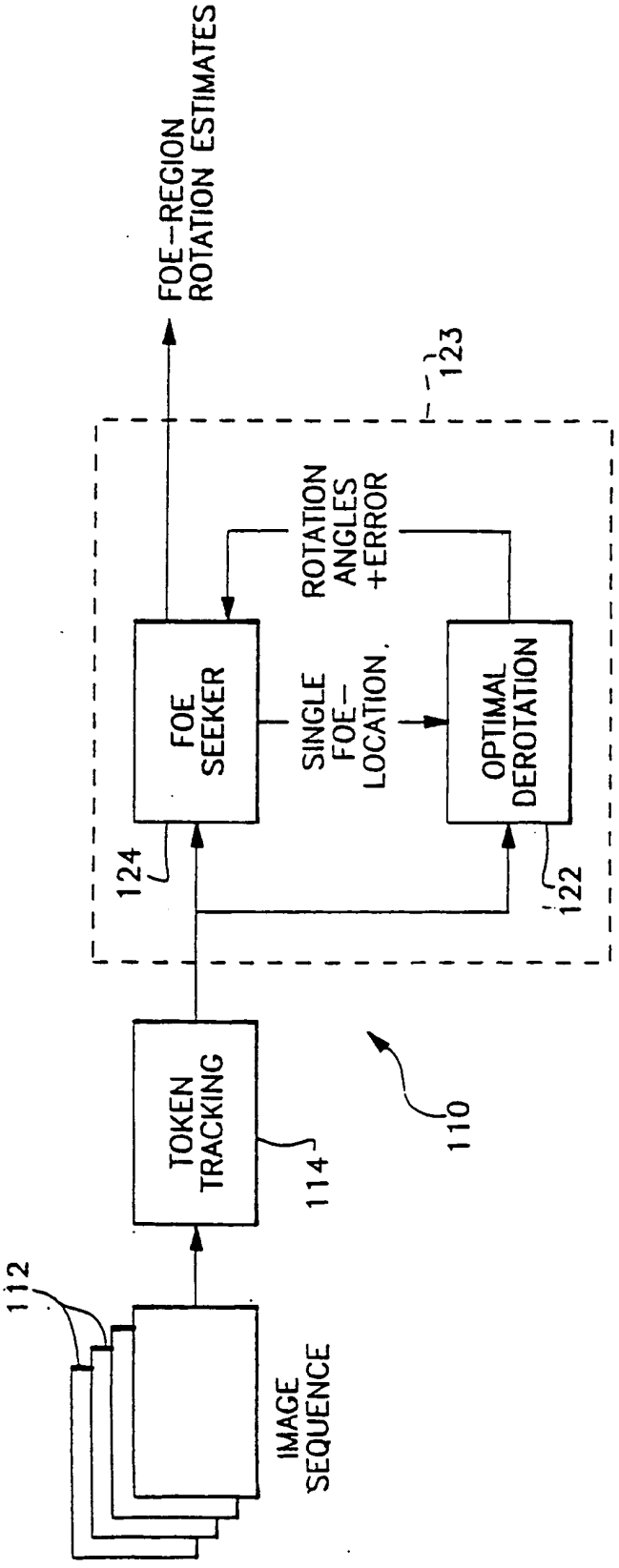

FOE-REGION ROTATION ESTIMATES
123
ROTATION ANGLES +ERROR
FOE SEEKER
SINGLE FOE-LOCATION
OPTIMAL DEROTATION
124
122
TOKEN TRACKING
114
110
112
IMAGE SEQUENCE

Fig. 1

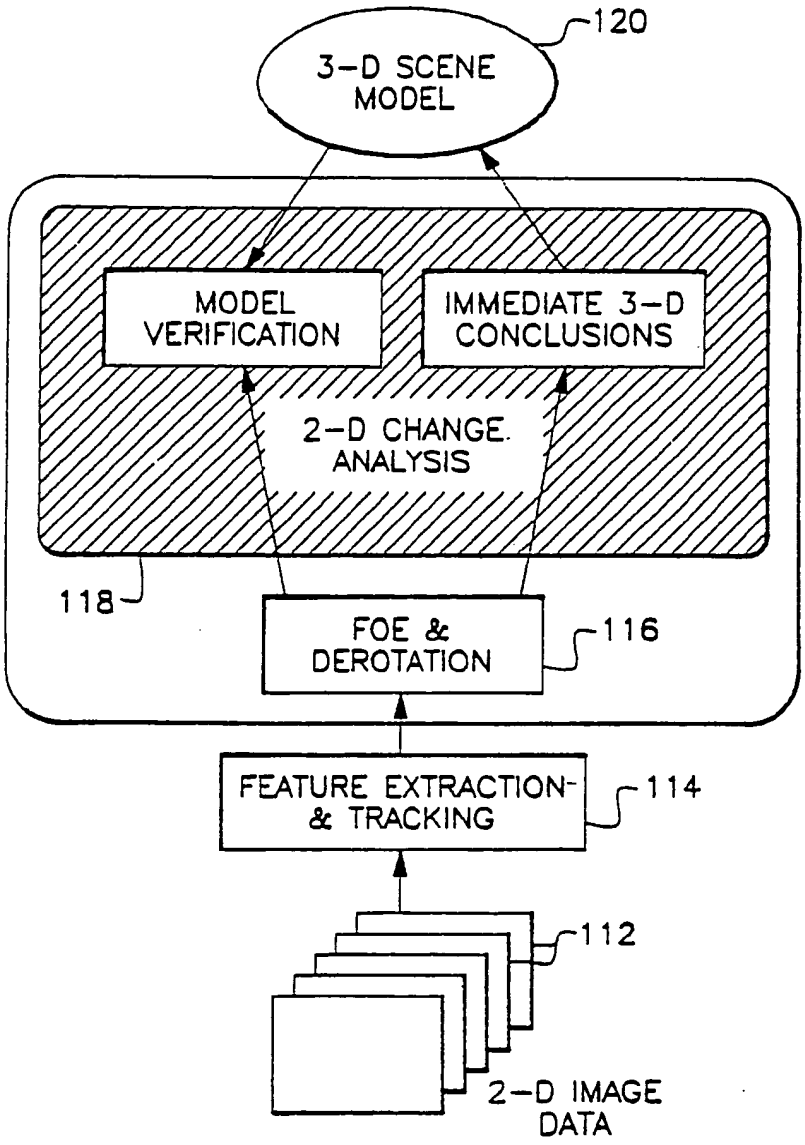

120
3-D SCENE MODEL
MODEL VERIFICATION
IMMEDIATE 3-D CONCLUSIONS
2-D CHANGE ANALYSIS
118
FOE & DEROTATION
116
FEATURE EXTRACTION & TRACKING
114
112
2-D IMAGE DATA

Fig. 2

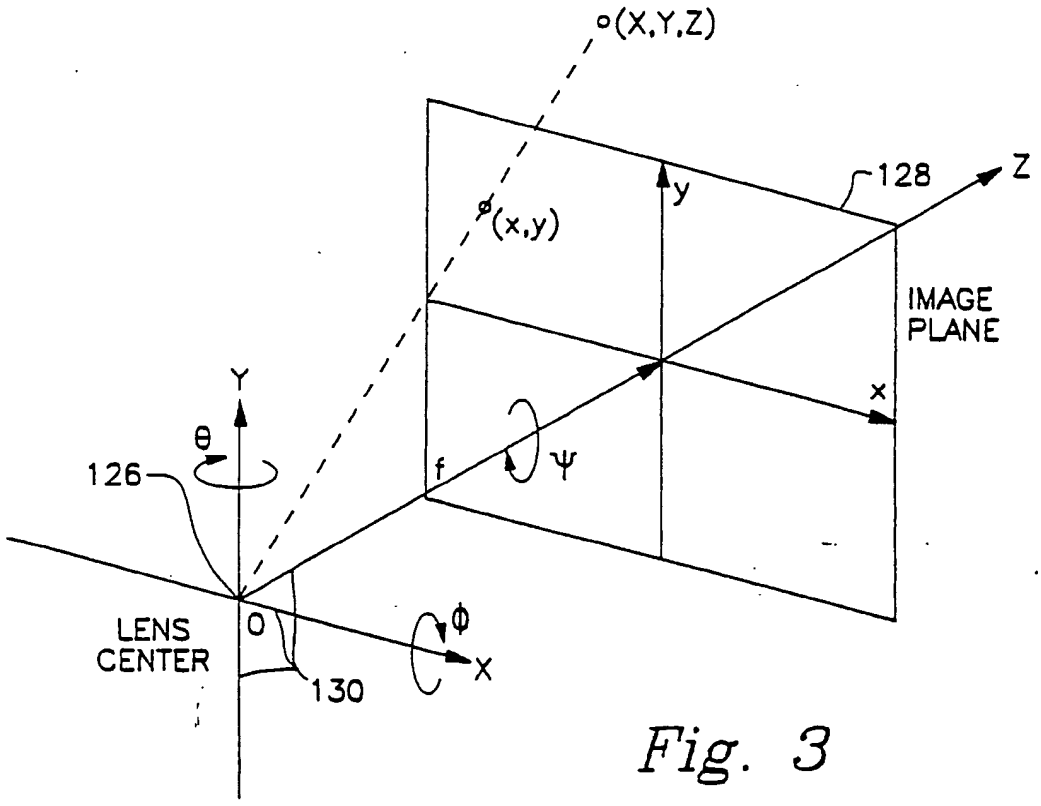


Fig. 3

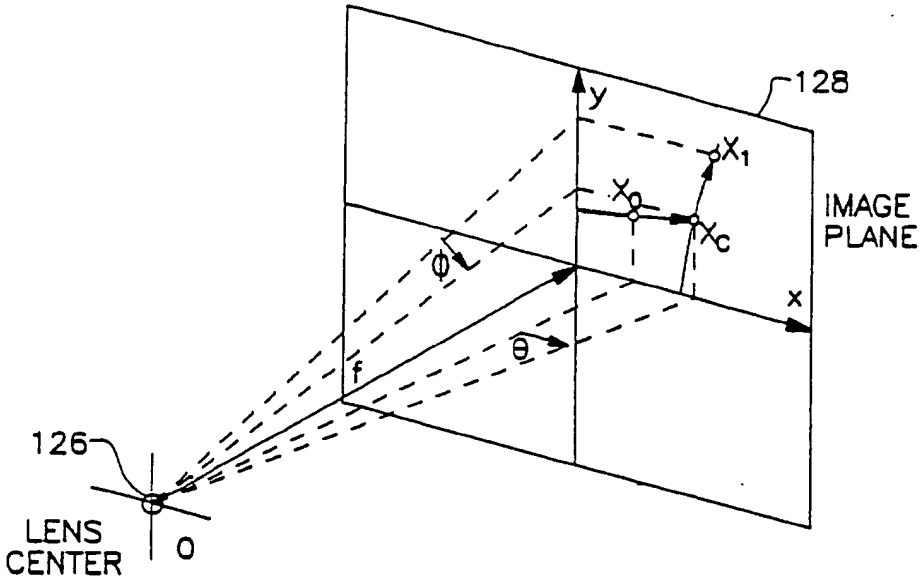


Fig. 4

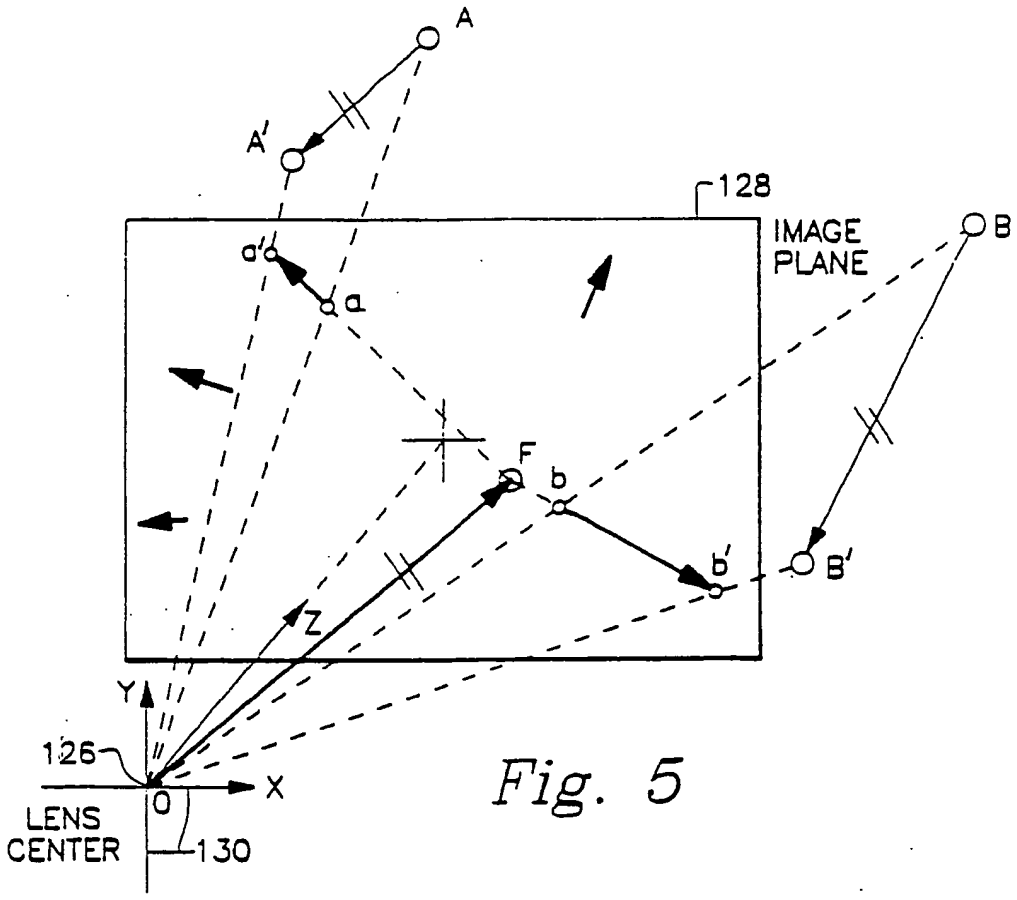


Fig. 5

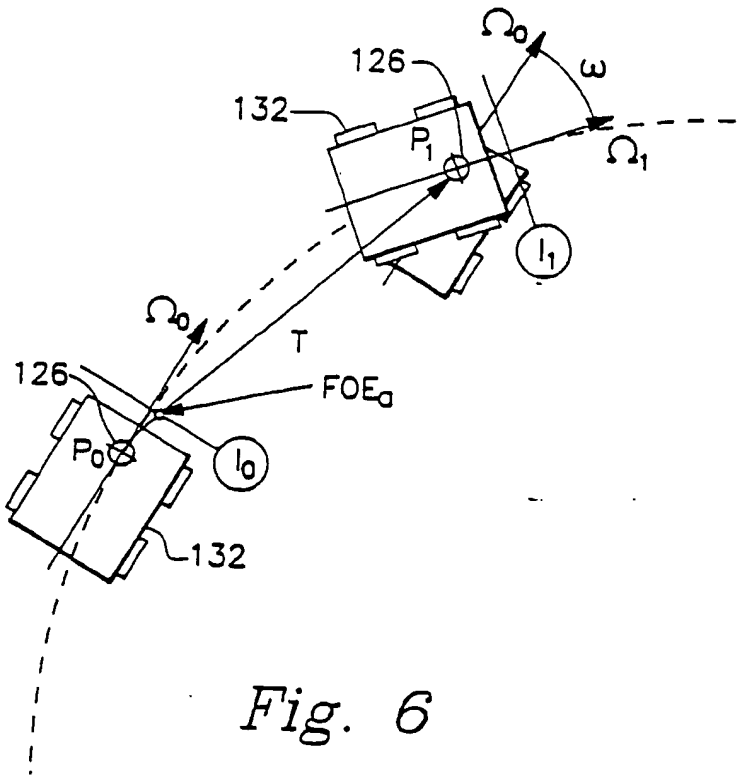


Fig. 6

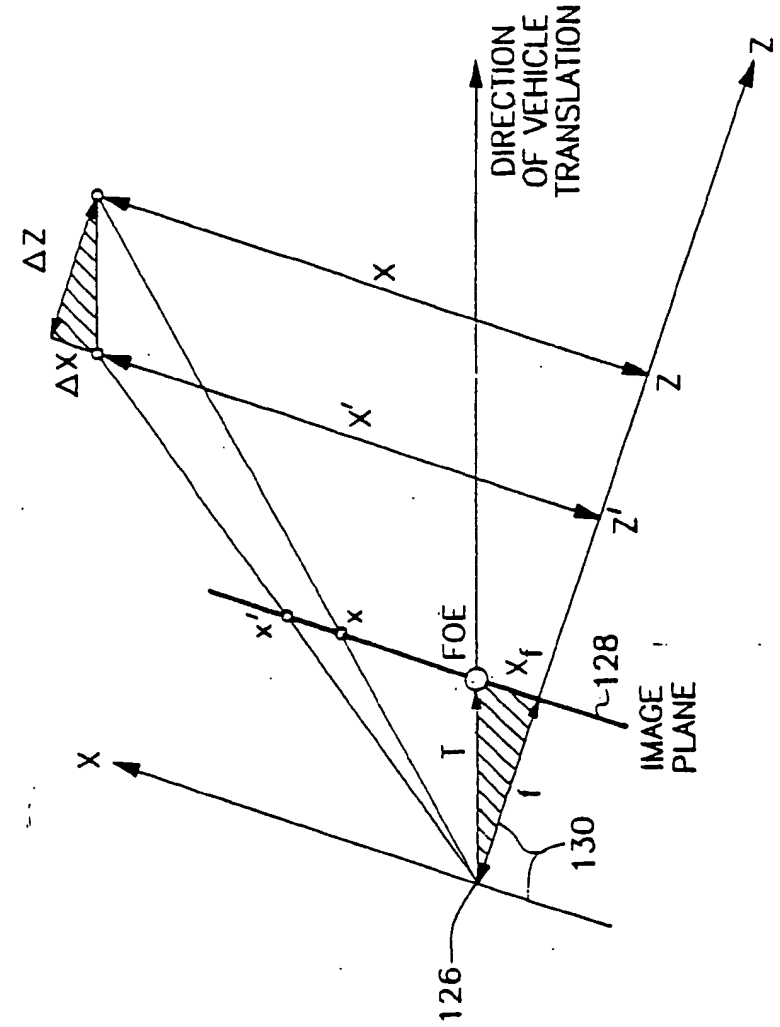

DIRECTION OF VEHICLE TRANSLATION
Z
ΔZ
ΔX
X
X'
Z
Z'
x'
x
FOE
x_f
128
IMAGE PLANE
T
f
130
126
X

Fig. 7

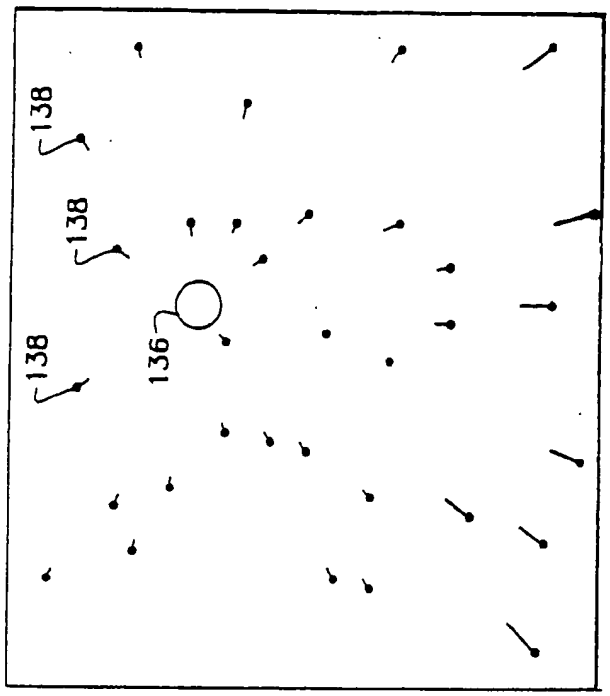


Fig. 8b

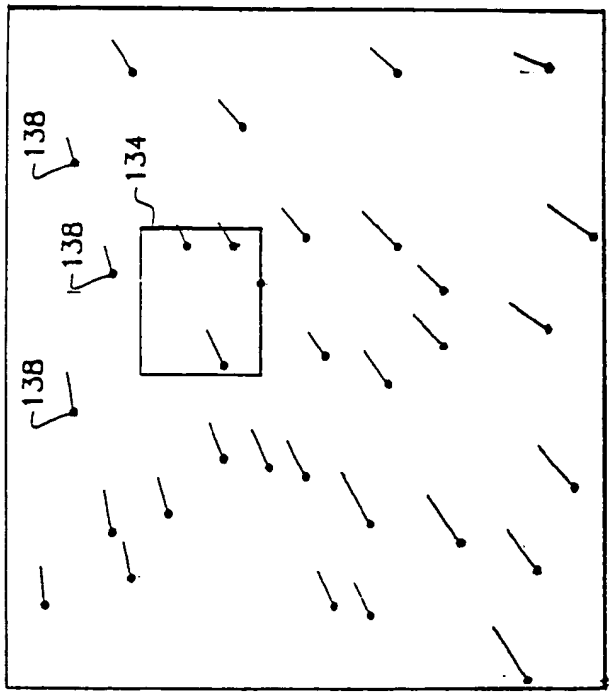


Fig. 8a

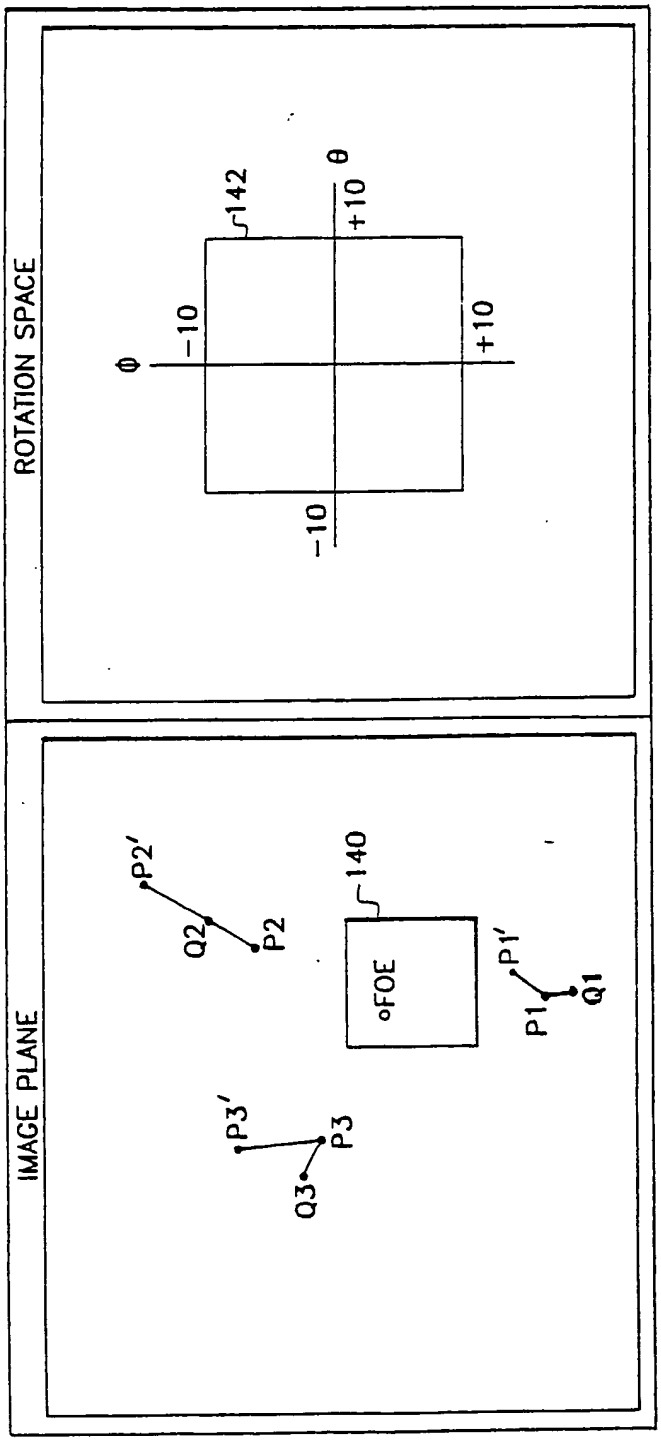

ROTATION SPACE
142
θ
+10
φ
-10
+10
-10
IMAGE PLANE
P2'
Q2
P2
140
oFOE
P1'
P1
Q1
P3'
P3
Q3

Fig. 9b

Fig. 9a

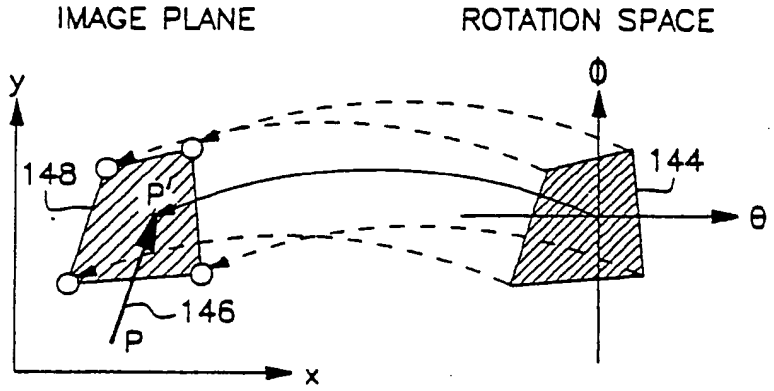


Fig. 10a

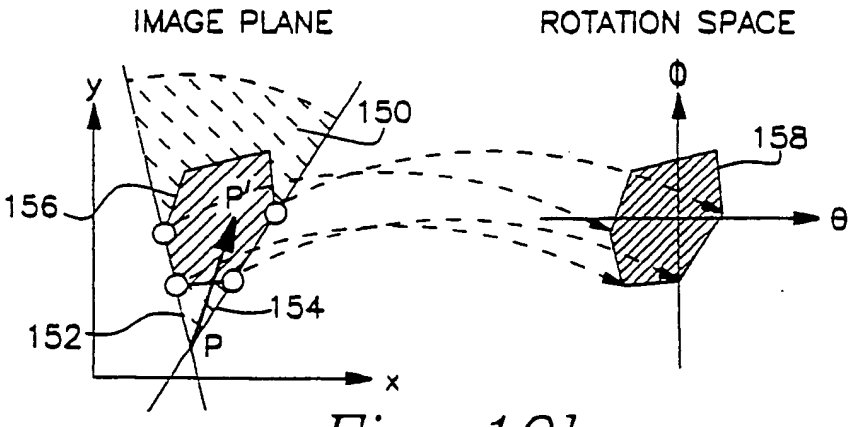


Fig. 10b

A 4112730

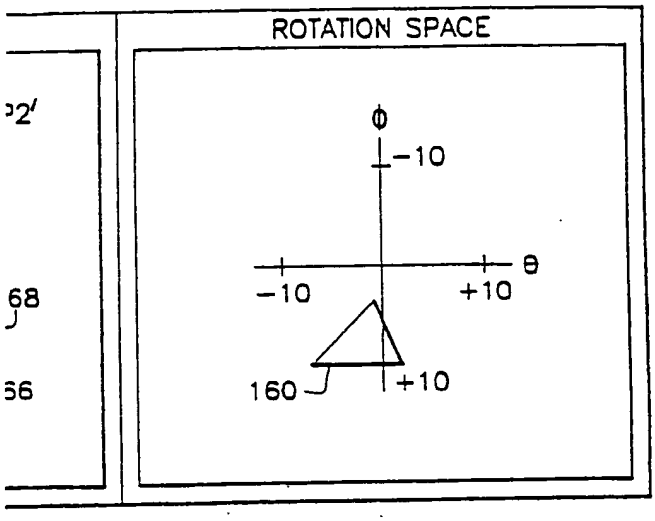


Fig. 11a

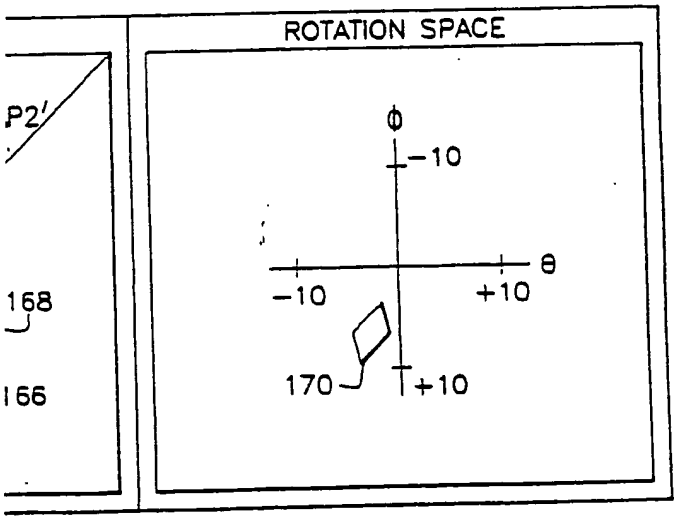


Fig. 11b

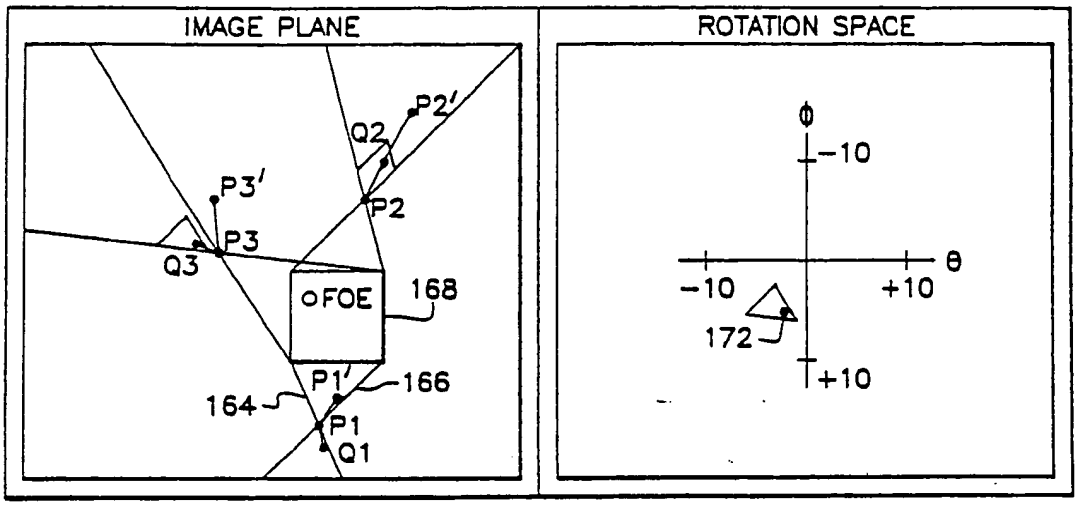


IMAGE PLANE
ROTATION SPACE
P2'
Q2
P2
P3'
P3
Q3
FOE
168
P1'
166
164
P1
Q1
φ
-10
θ
-10
+10
172
+10

Fig. 11c

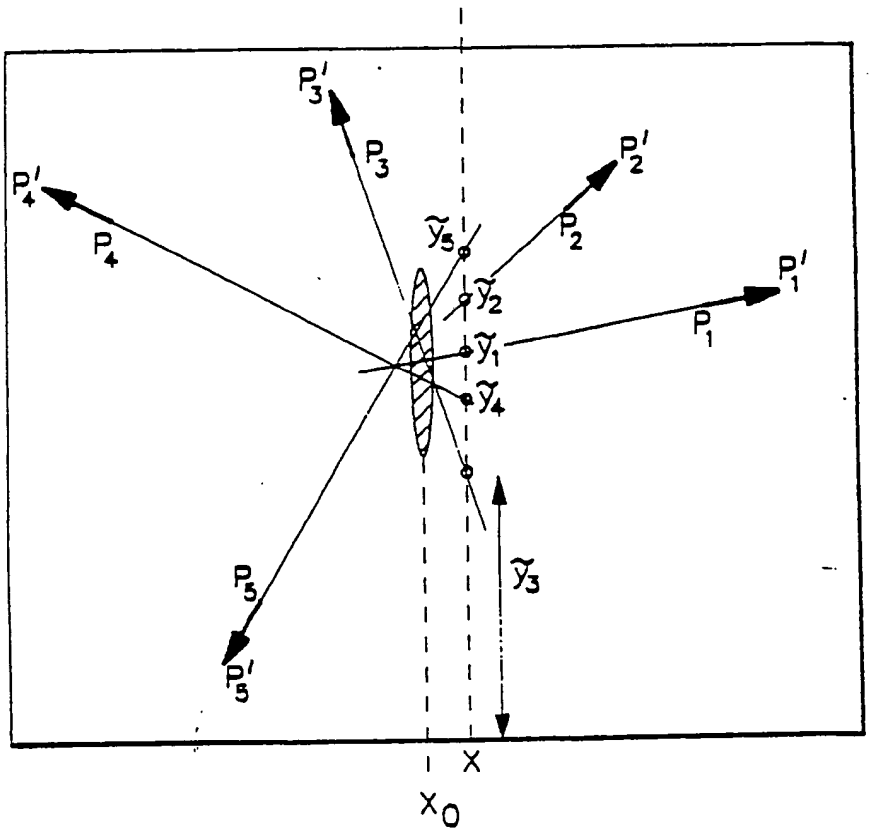


Fig. 12

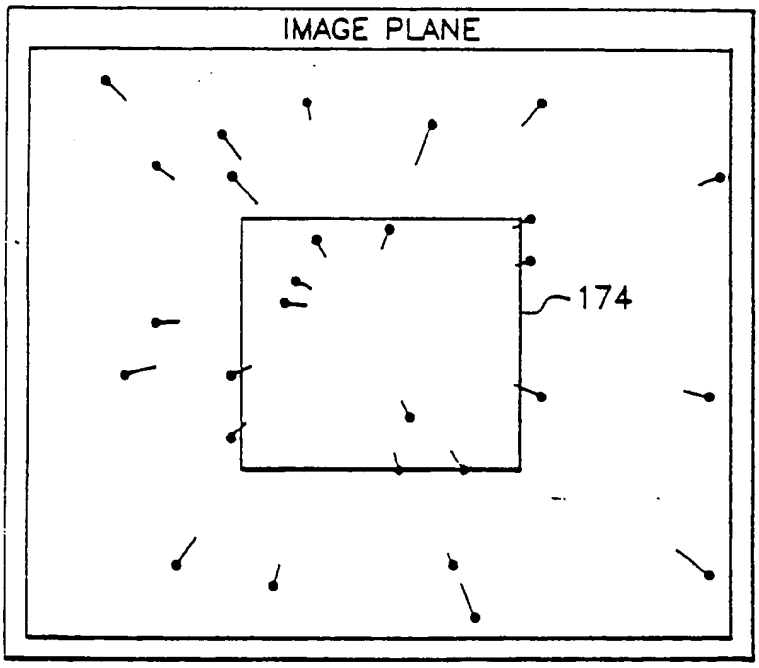
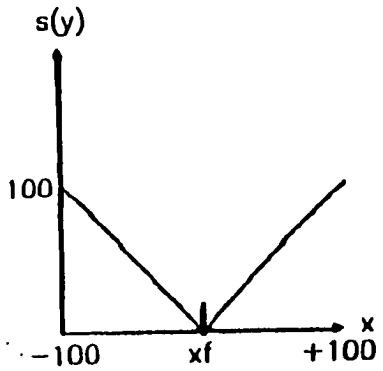
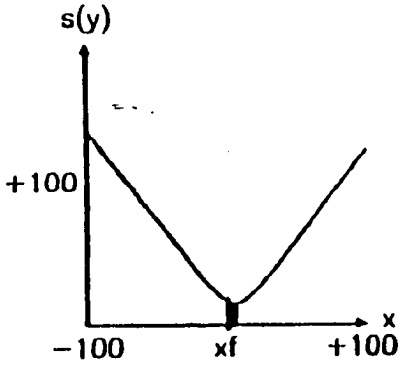


Fig. 13



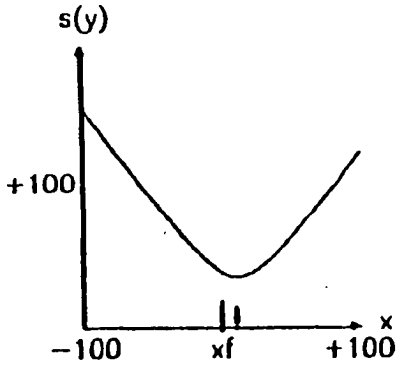
ROTATION ERROR:
0.0/0.0 DEG.

Fig. 14a



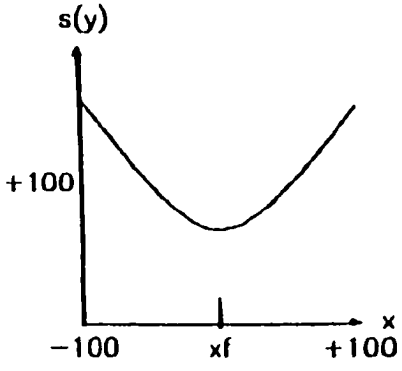
ROTATION ERROR:
0.0/0.2 DEG

Fig. 14b



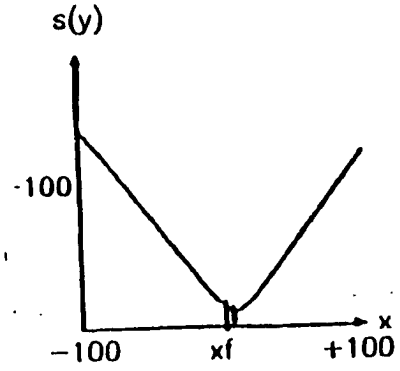
ROTATION ERROR:
0.0/0.5 DEG

Fig. 14c



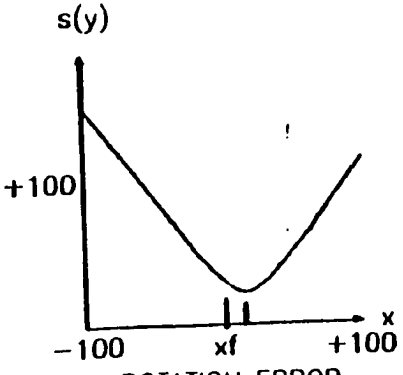
ROTATION ERROR:
0.0/1.0 DEG

Fig. 14d



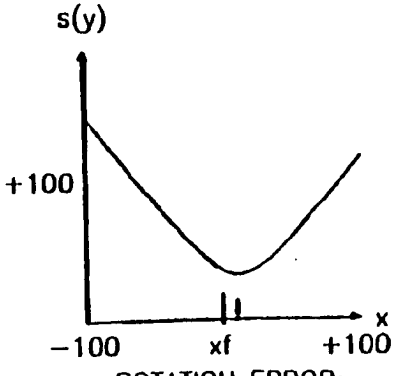
ROTATION ERROR:
0.0/0.0 DEG

Fig. 15a



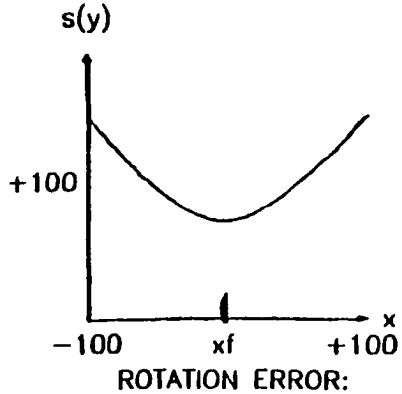
ROTATION ERROR:
0.0/0.2 DEG

Fig. 15b



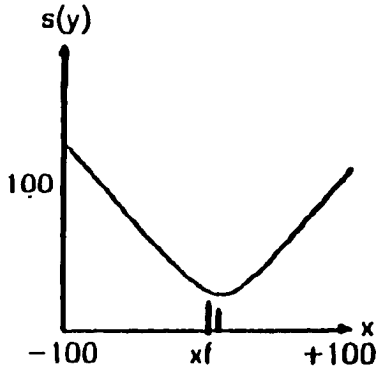
ROTATION ERROR:
0.0/0.5 DEG

Fig. 15c



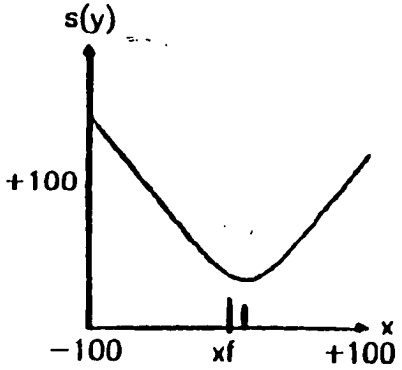
ROTATION ERROR:
0.0/1.0 DEG

Fig. 15d



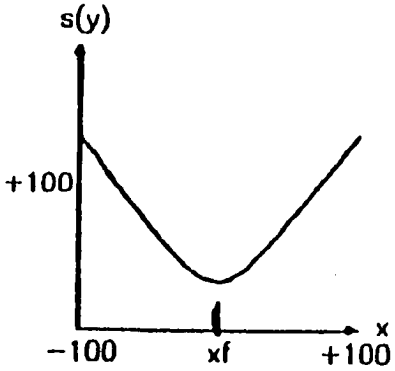
ROTATION ERROR:
0.0/0.0 DEG.

Fig. 16a



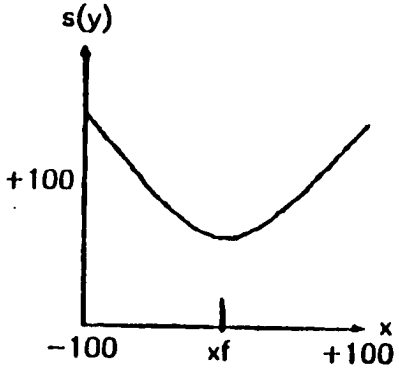
ROTATION ERROR:
0.0/0.2 DEG

Fig. 16b



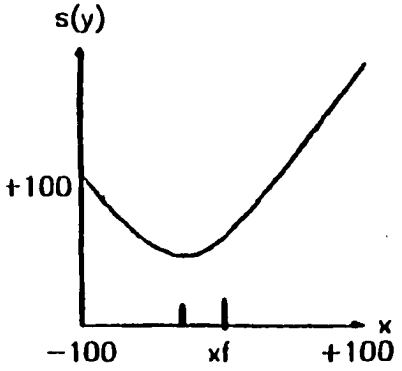
ROTATION ERROR:
0.0/0.5 DEG

Fig. 16c



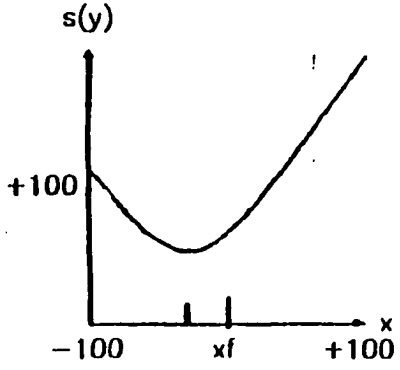
ROTATION ERROR:
0.0/1.0 DEG

Fig. 16d



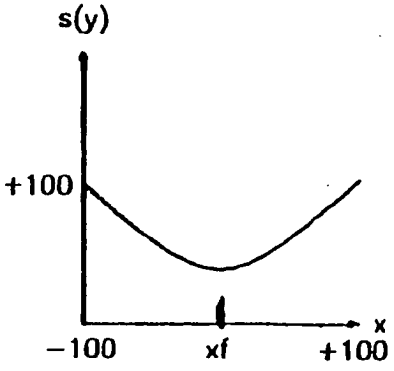
ROTATION ERROR:
0.0/0.0 DEG

Fig. 17a



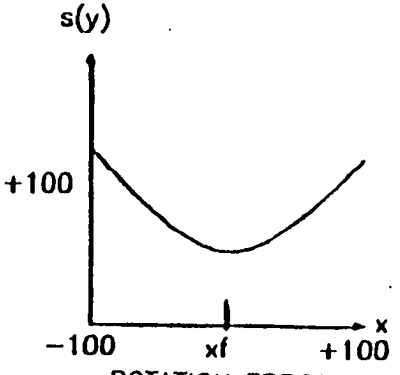
ROTATION ERROR:
0.0/0.2 DEG

Fig. 17b



ROTATION ERROR:
0.0/0.5 DEG

Fig. 17c



ROTATION ERROR:
0.0/1.0 DEG

Fig. 17d

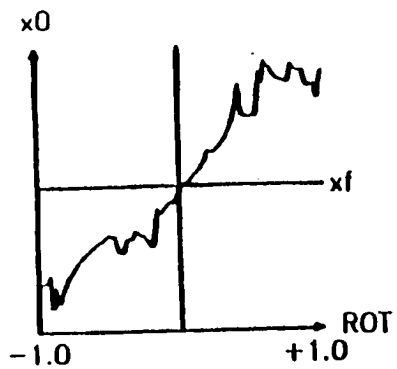


Fig. 18a

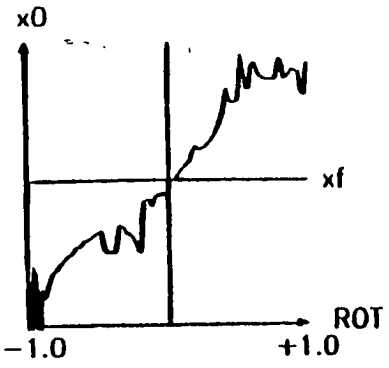


Fig. 18b

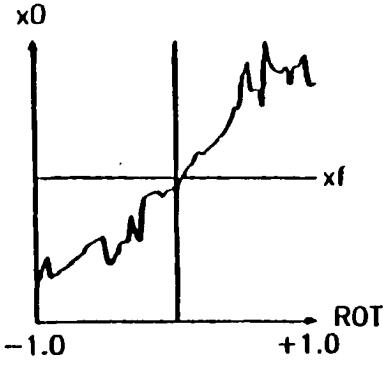


Fig. 18c

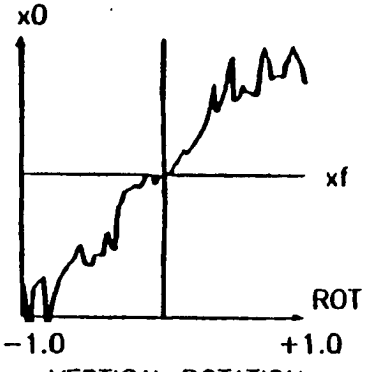


Fig. 18d

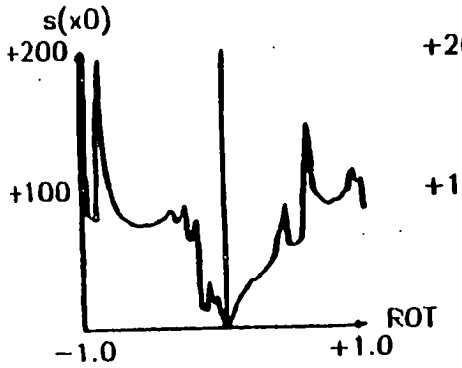


Fig. 19a

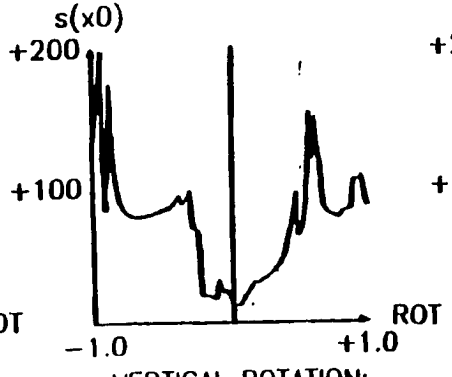


Fig. 19b

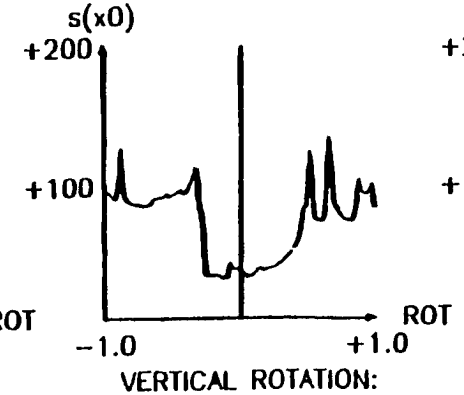


Fig. 19c

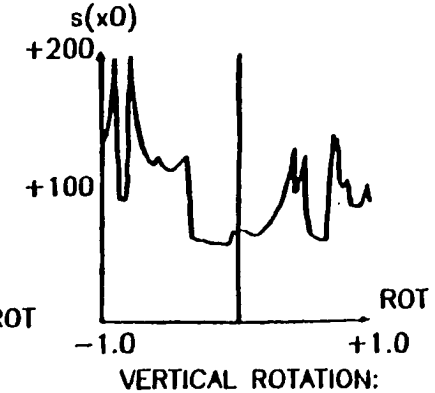


Fig. 19d

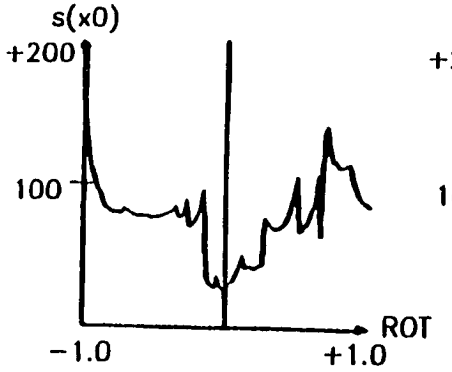


Fig. 20a

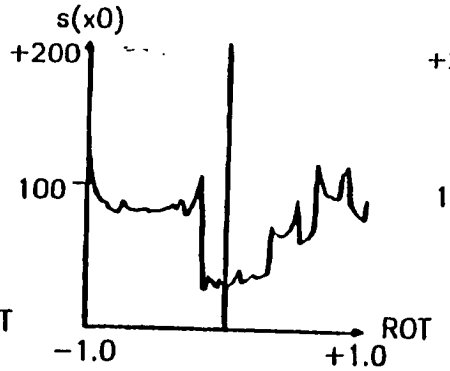


Fig. 20b

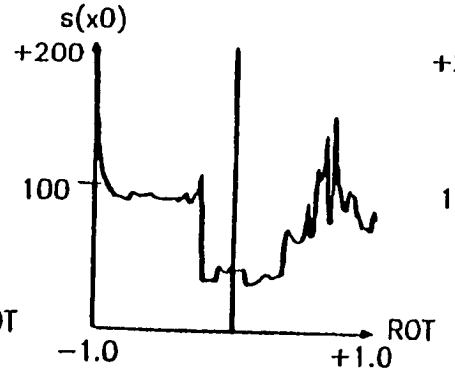


Fig. 20c

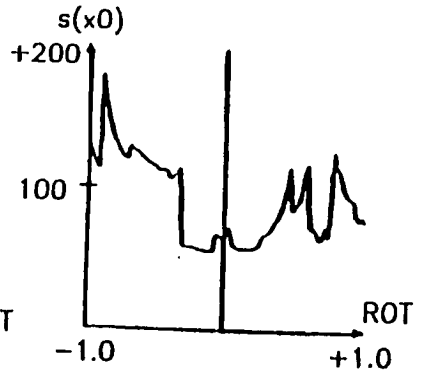


Fig. 20d

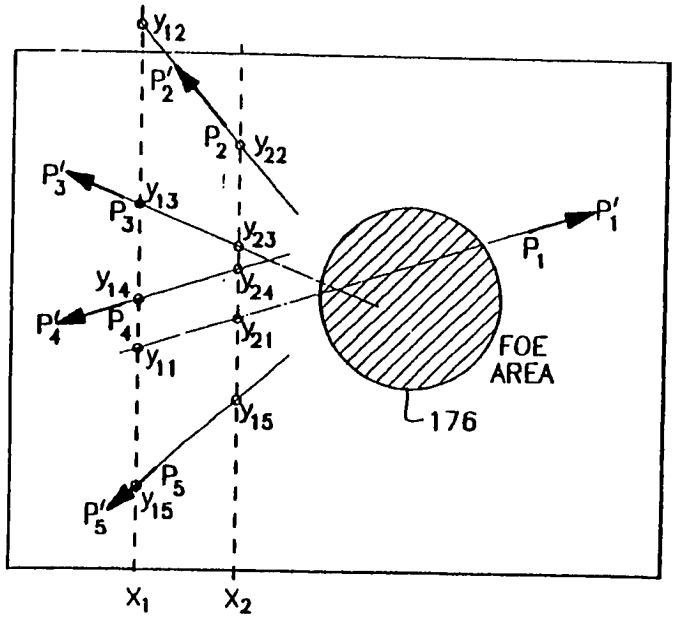


Fig. 21

EP 0 390 051 A2

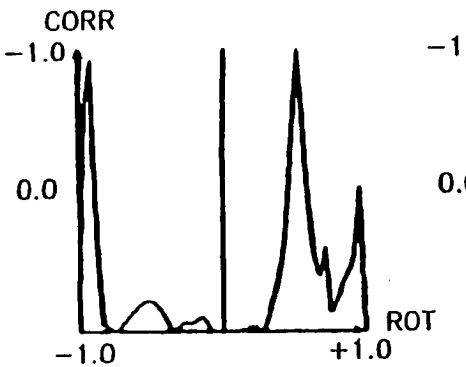


Fig. 22a

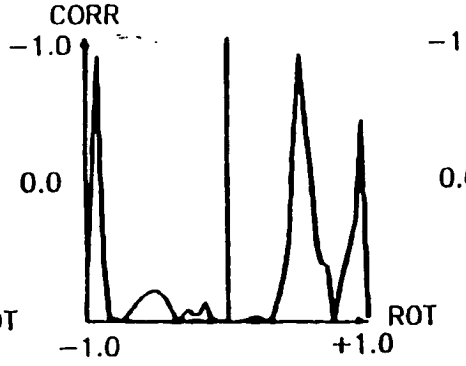


Fig. 22b

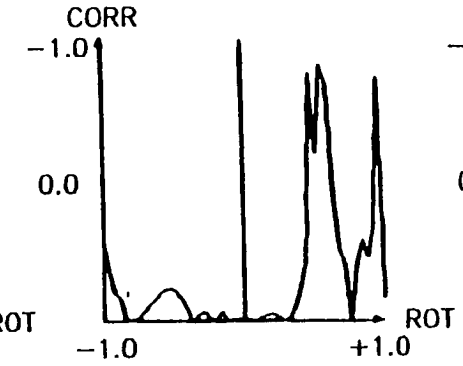


Fig. 22c

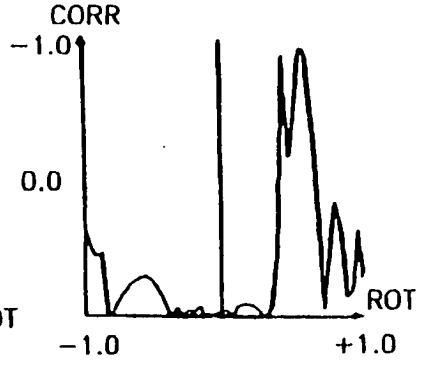


Fig. 22d

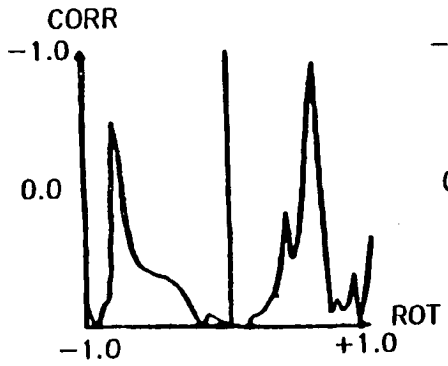


Fig. 23a

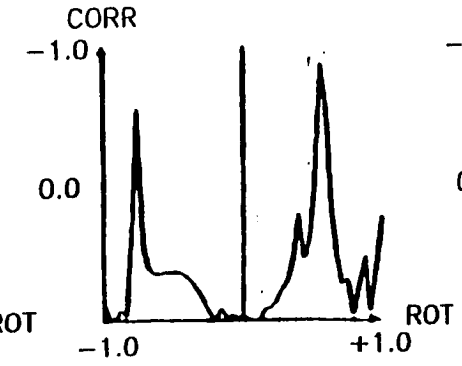


Fig. 23b

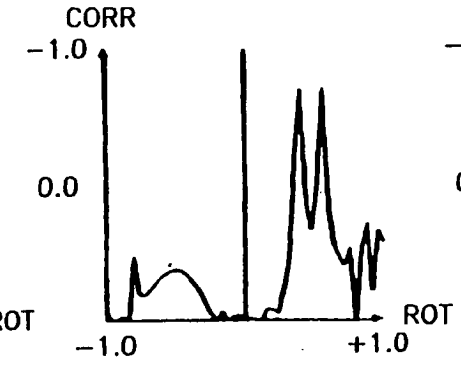


Fig. 23c

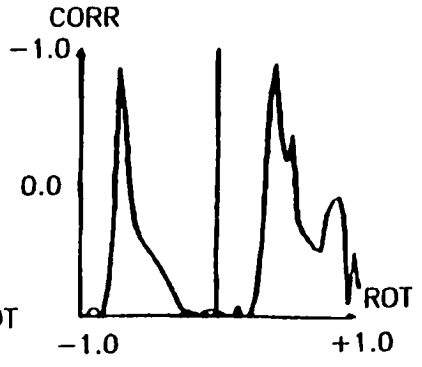


Fig. 23d

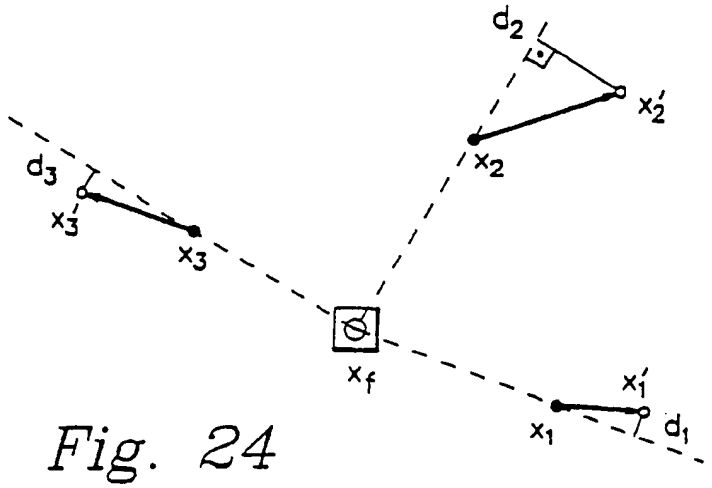


Fig. 24

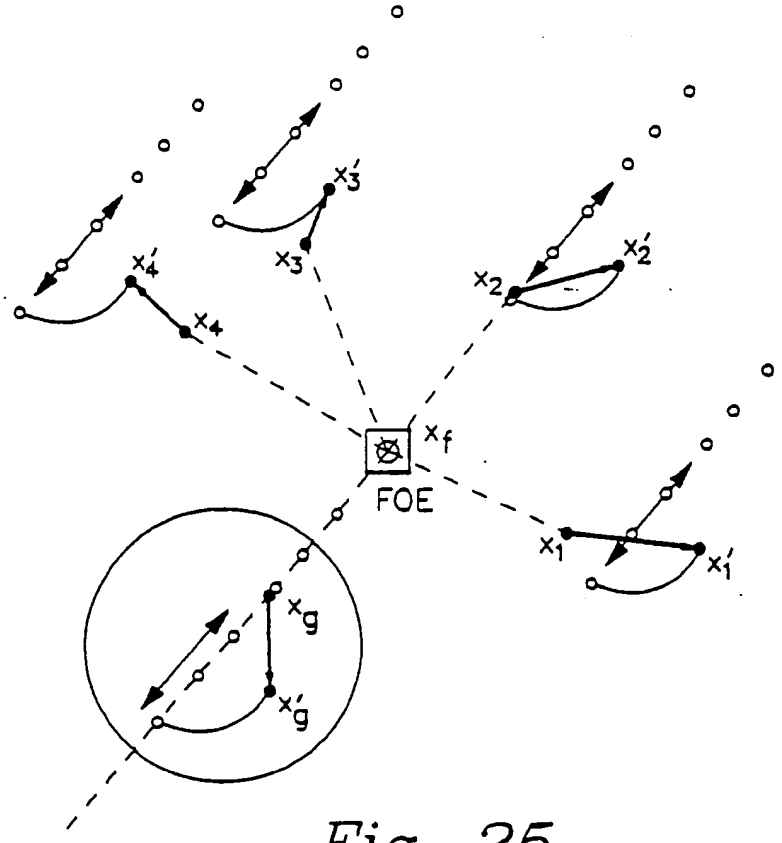


Fig. 25

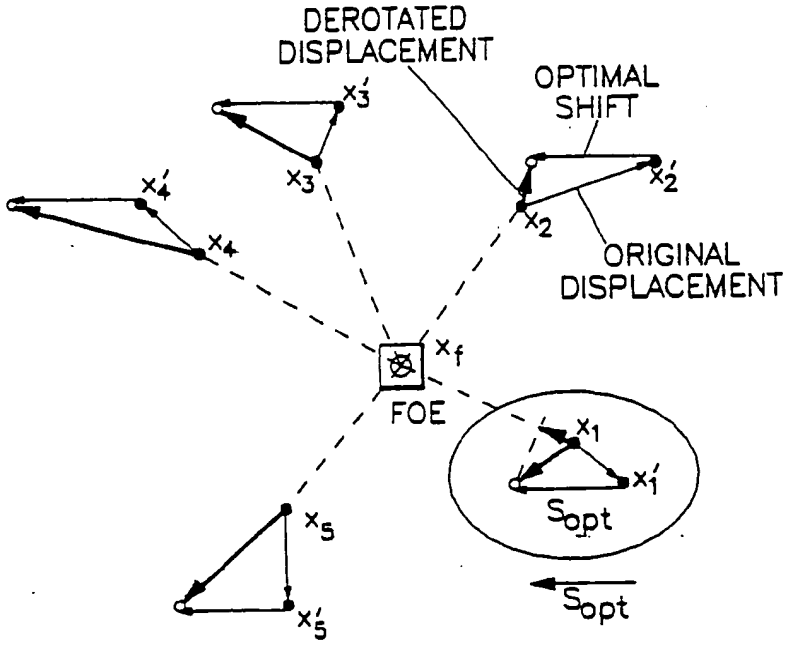

DEROTATED
DISPLACEMENT
OPTIMAL
SHIFT
ORIGINAL
DISPLACEMENT
FOE
x_f
x_1
x_1'
x_2
x_2'
x_3
x_3'
x_4
x_4'
x_5
x_5'
S_{opt}
S_{opt}

Fig. 26

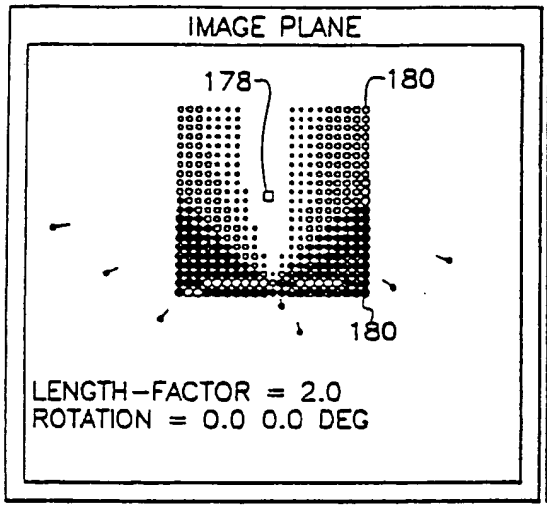


Fig. 27a

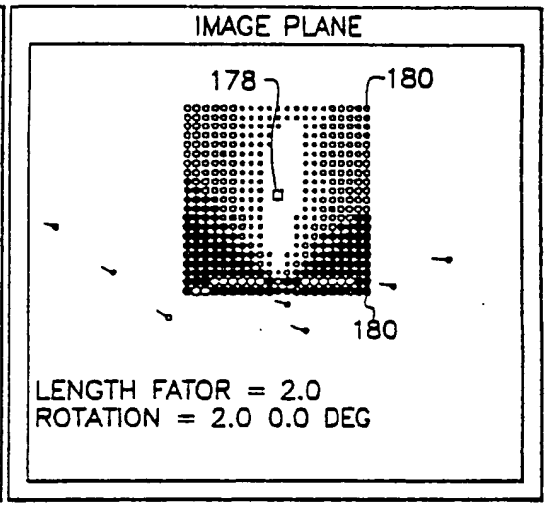


Fig. 27b

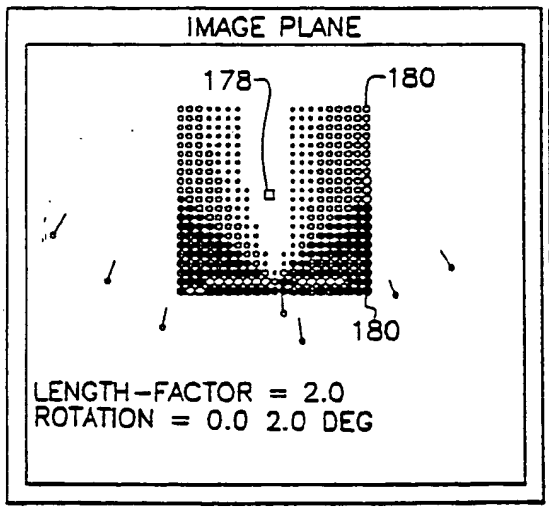


Fig. 27c

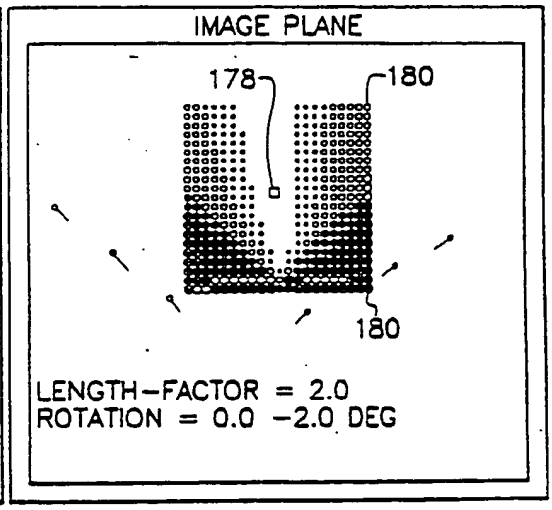
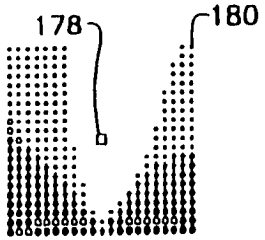
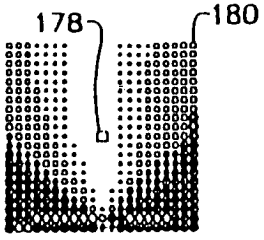


Fig. 27d



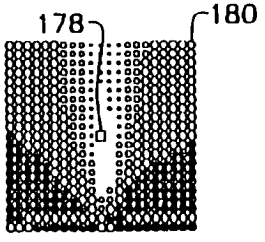
LENGTH–FACTOR = 1.0
ROTATION = 0.0 0.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

Fig. 28a



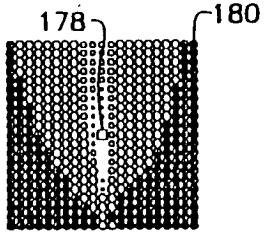
LENGTH–FACTOR = 2.0
ROTATION = 0.0 0.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

Fig. 28b



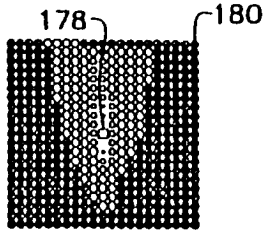
LENGTH–FACTOR = 5.0
ROTATION = 0.0 0.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

Fig. 28c



LENGTH–FACTOR = 10.0
ROTATION = 0.0 0.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

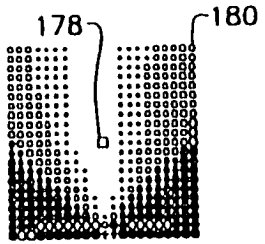
Fig. 28d



LENGTH–FACTOR = 15.0
ROTATION = 0.0 0.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

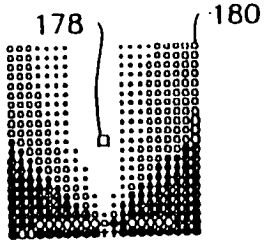
Fig. 28e

ERROR < 1.0 · ERROR = 1.0 ◦ ERROR = 2.0 ◦ ERROR = 3.0 ◦ ERROR = 4.0 ◦ ERROR > 4.0 · PROHIBITED



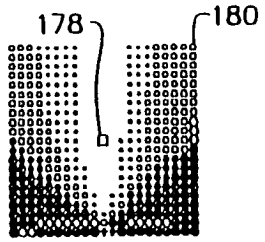
LENGTH–FACTOR = 2.0
ROTATION = 0.0 0.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

Fig. 29a



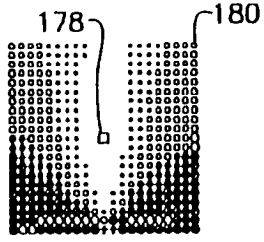
LENGTH–FACTOR = 2.0
ROTATION = 0.1 0.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

Fig. 29b



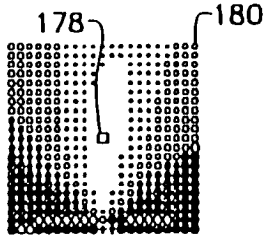
LENGTH–FACTOR = 2.0
ROTATION = 0.5 0.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

Fig. 29c



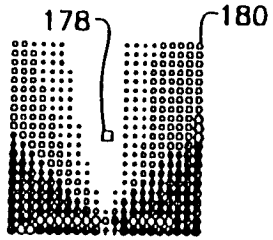
LENGTH–FACTOR = 2.0
ROTATION = 1.0 0.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

Fig. 29d



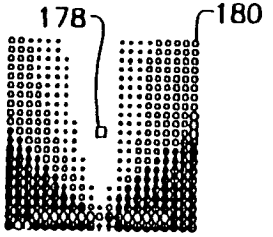
LENGTH–FACTOR = 2.0
ROTATION = 2.0 0.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

Fig. 29e



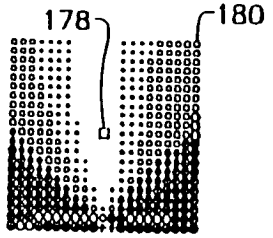
LENGTH–FACTOR = 2.0
ROTATION = 0.0 –2.0 DEG
NOISE = ± 0.0 PXLS UNIFORM

Fig. 30a



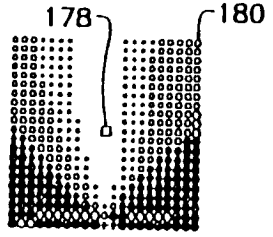
LENGTH–FACTOR = 2.0
ROTATION = 0.0 –1.0 DEG
NOISE = ± 0.0 PXLS UNIFORM

Fig. 30b



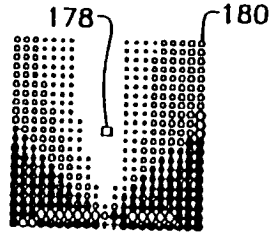
LENGTH–FACTOR = 2.0
ROTATION = 0.0 0.0 DEG
NOISE = ± 0.0 PXLS UNIFORM

Fig. 30c



LENGTH–FACTOR = 2.0
ROTATION = 0.0 1.0 DEG
NOISE = ± 0.0 PXLS UNIFORM

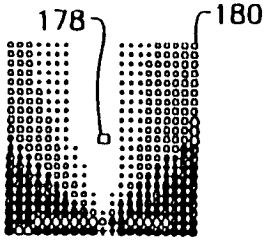
Fig. 30d



LENGTH–FACTOR = 2.0
ROTATION = 0.0 2.0 DEG
NOISE = ± 0.0 PXLS UNIFORM

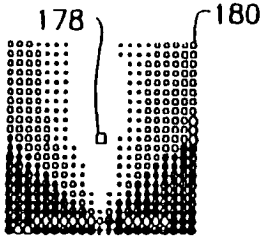
Fig. 30e

ERROR < 1.0 · ERROR = 1.0 ◦ ERROR = 2.0 ◦ ERROR = 3.0 ◦ ERROR = 4.0 ◦ ERROR > 4.0 · PROHIBITED



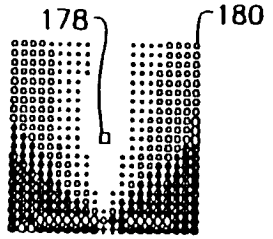
LENGTH–FACTOR = 2.0
ROTATION = 0.0 0.0 DEG
NOISE = ± 0.0 PXLS UNIFORM

Fig. 31a



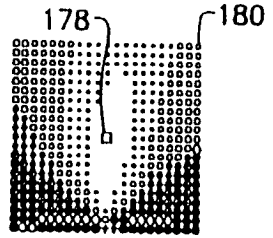
LENGTH–FACTOR = 2.0
ROTATION = 0.5 0.5 DEG
NOISE = ± 0.0 PXLS UNIFORM

Fig. 31b



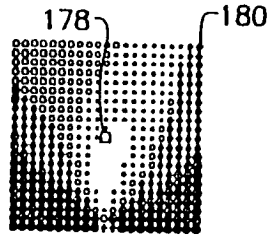
LENGTH–FACTOR = 2.0
ROTATION = 1.0 1.0 DEG
NOISE = ± 0.0 PXLS UNIFORM

Fig. 31c



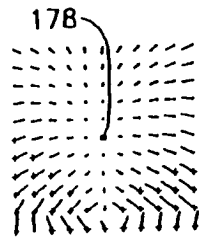
LENGTH–FACTOR = 2.0
ROTATION = 2.0 2.0 DEG
NOISE = ± 0.0 PXLS UNIFORM

Fig. 31d



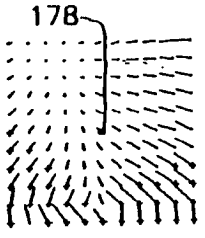
LENGTH–FACTOR = 2.0
ROTATION = 4.0 4.0 DEG
NOISE = ± 0.0 PXLS UNIFORM

Fig. 31e



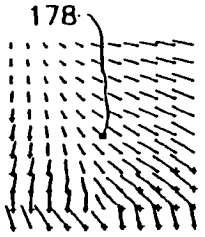
LENGTH–FACTOR = 2.0
ROTATION = 0.0 0.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

Fig. 31f



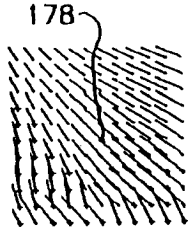
LENGTH–FACTOR = 2.0
ROTATION = 0.5 0.5 DEG
NOISE = ± 0.0
PXLS UNIFORM

Fig. 31g



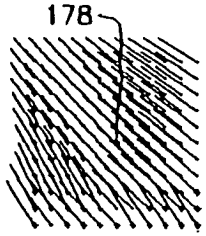
LENGTH–FACTOR = 2.0
ROTATION = 1.0 1.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

Fig. 31h



LENGTH–FACTOR = 2.0
ROTATION = 2.0 2.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

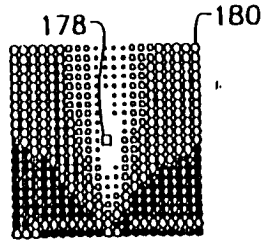
Fig. 31i



LENGTH–FACTOR = 2.0
ROTATION = 4.0 4.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

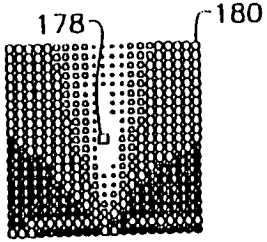
Fig. 31j

ERROR < 1.0 · ERROR = 1.0 ◦ ERROR = 2.0 ◦ ERROR = 3.0 ◦ ERROR = 4.0 ◦ ERROR > 4.0 • PROHIBITED



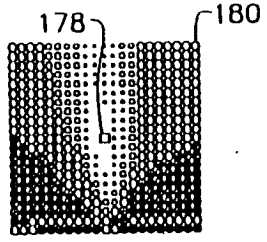
LENGTH–FACTOR = 5.0
ROTATION = 0.0 0.0 DEG
NOISE = ± 0.0
PXLS UNIFORM

Fig. 32a



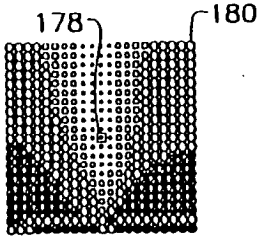
LENGTH–FACTOR = 5.0
ROTATION = 0.0 0.0 DEG
NOISE = ±0.5
PXLS UNIFORM

Fig. 32b



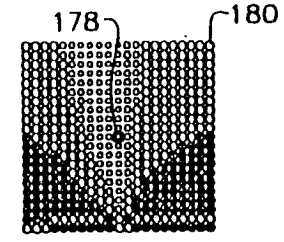
LENGTH–FACTOR = 5.0
ROTATION = 0.0 0.0 DEG
NOISE = ± 1.0
PXLS UNIFORM

Fig.32c



LENGTH–FACTOR = 5.0
ROTATION = 0.0 0.0 DEG
NOISE = ± 2.0
PXLS UNIFORM

Fig. 32d



LENGTH–FACTOR = 5.0
ROTATION = 0.0 0.0 DEG
NOISE = ±4.0
PXLS UNIFORM

Fig.32e

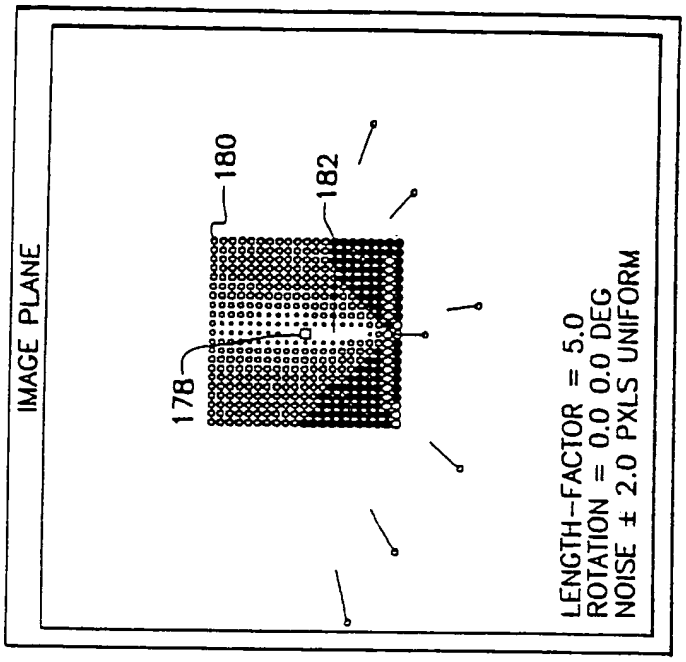


Fig. 33b

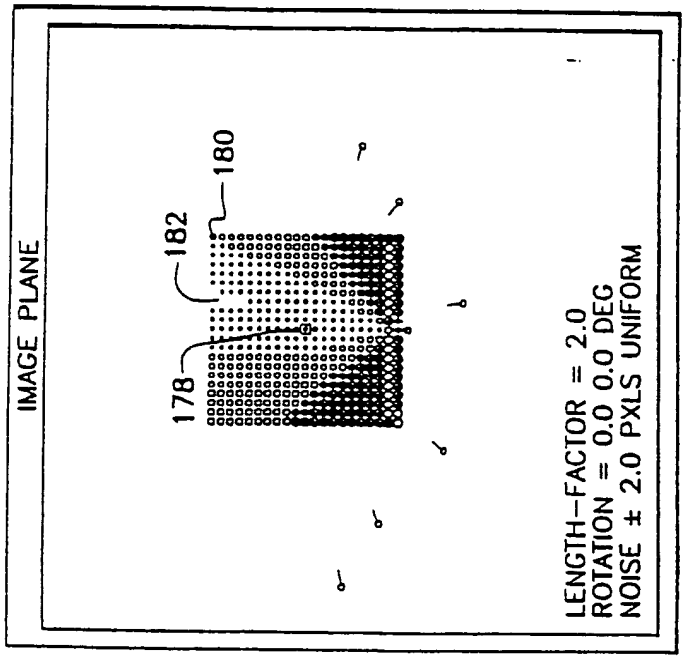


Fig. 33a

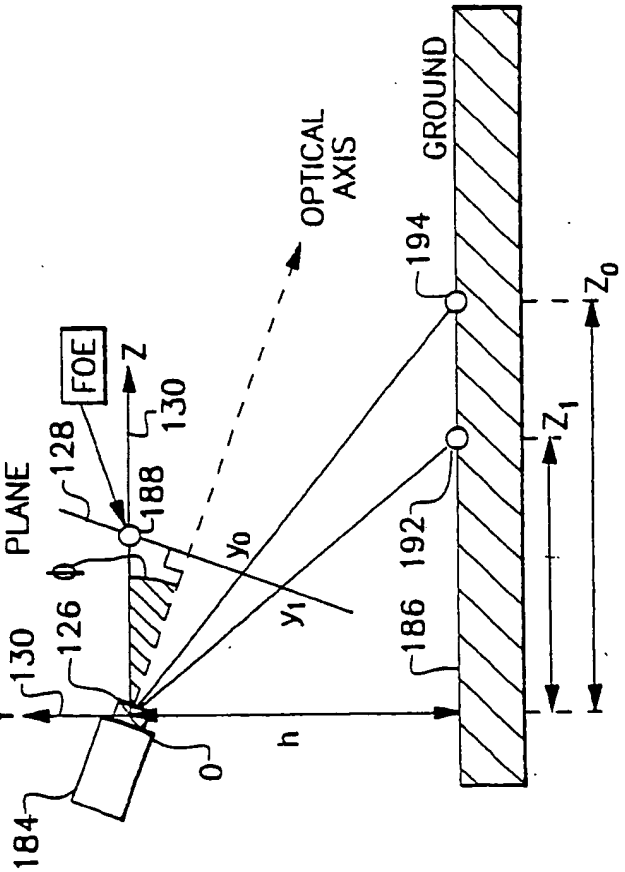

OPTICAL AXIS
GROUND
194
Z_0
Z_1
FOE
Z
130
128
188
IMAGE PLANE
y_0
y_1
192
186
126
130
Y
0
h
184

Fig. 34

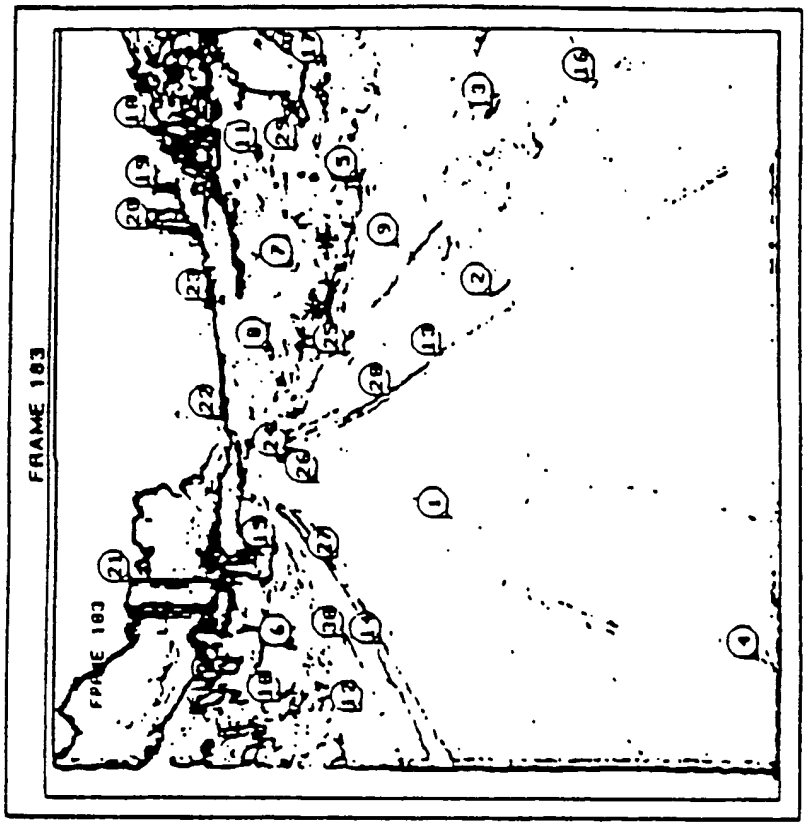


Fig. 35b

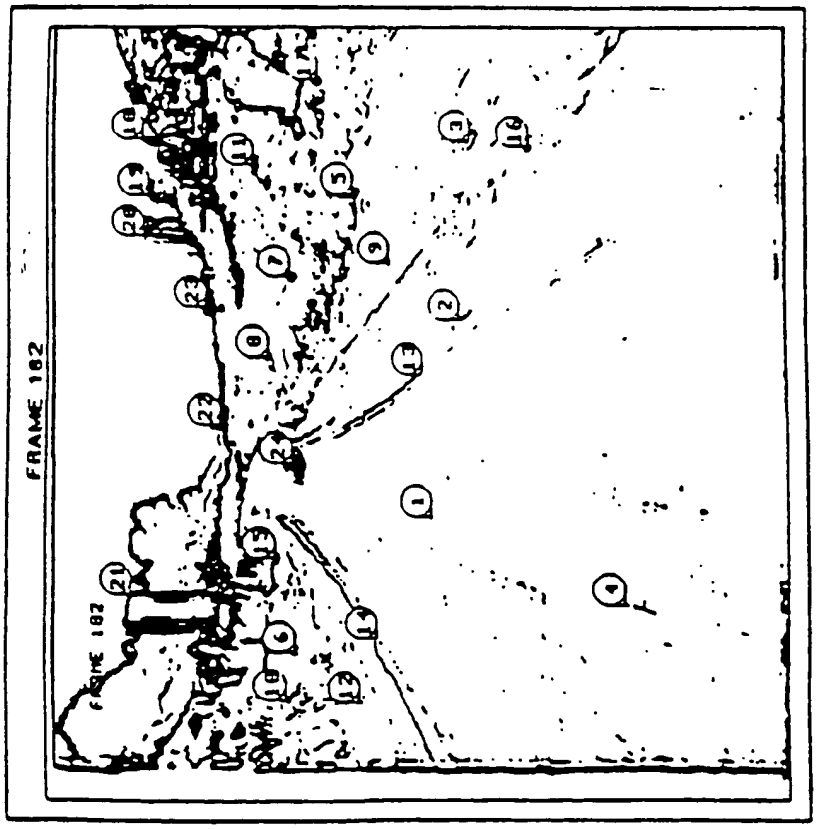


Fig. 35a

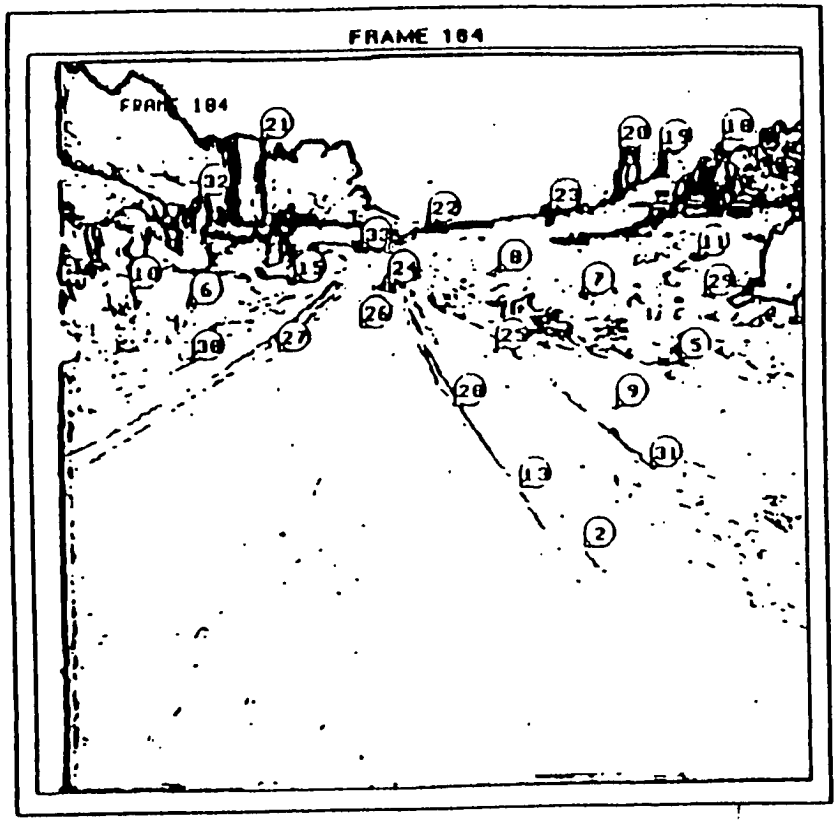


Fig. 35c

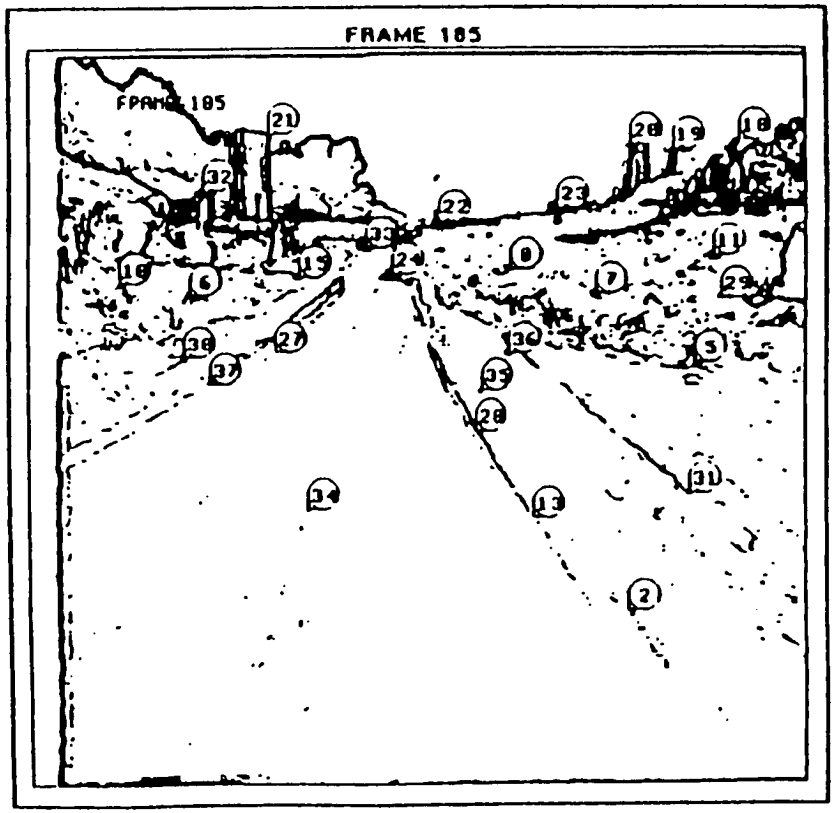


Fig. 35d

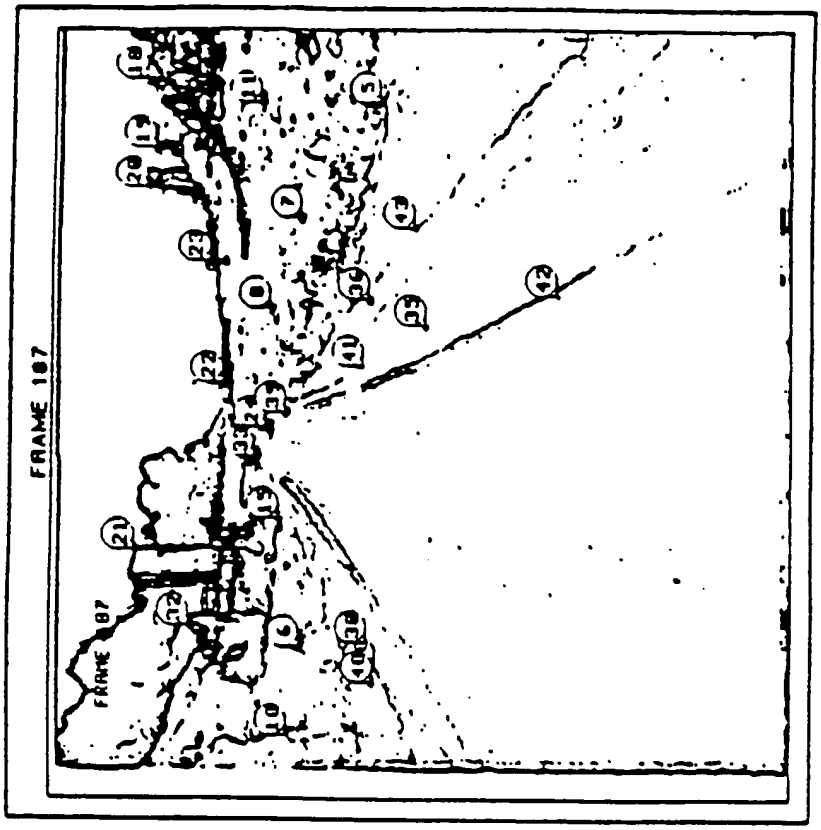


Fig. 35f

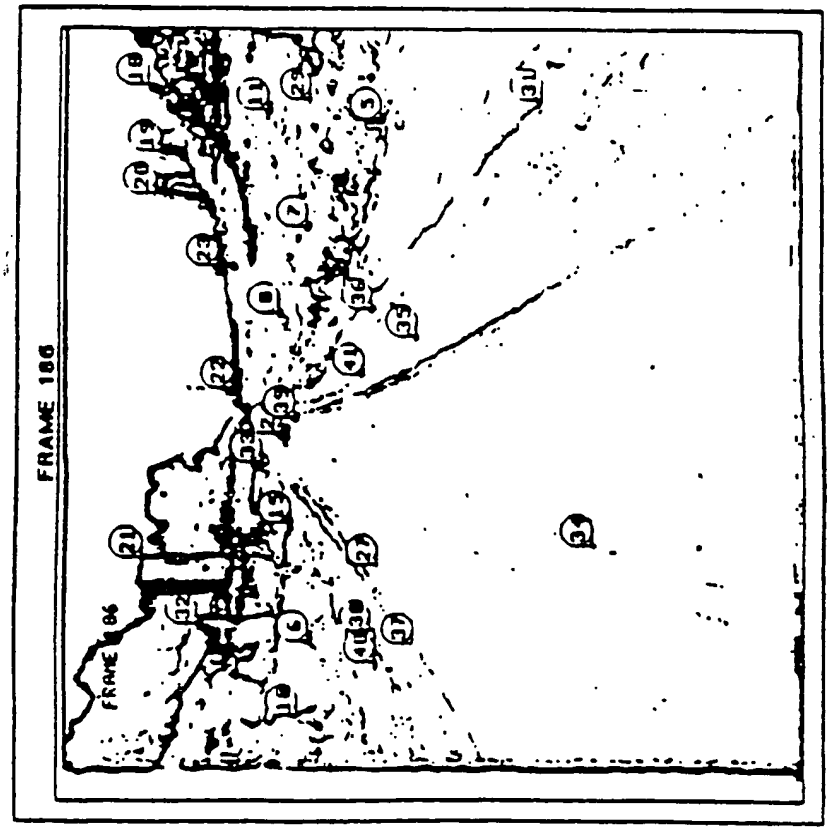


Fig. 35e

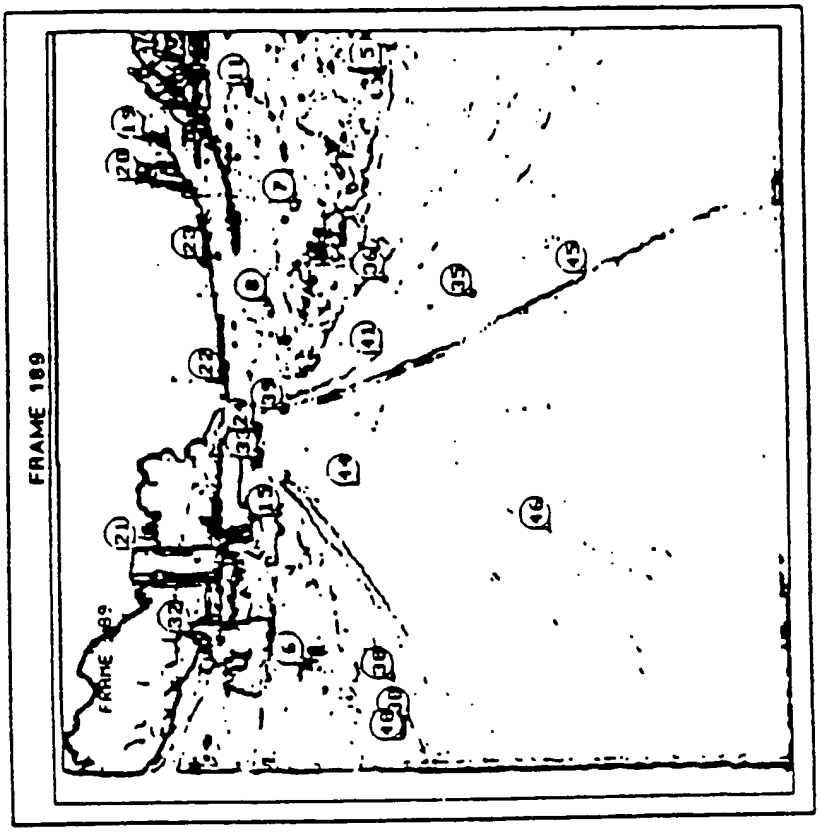


Fig. 35h

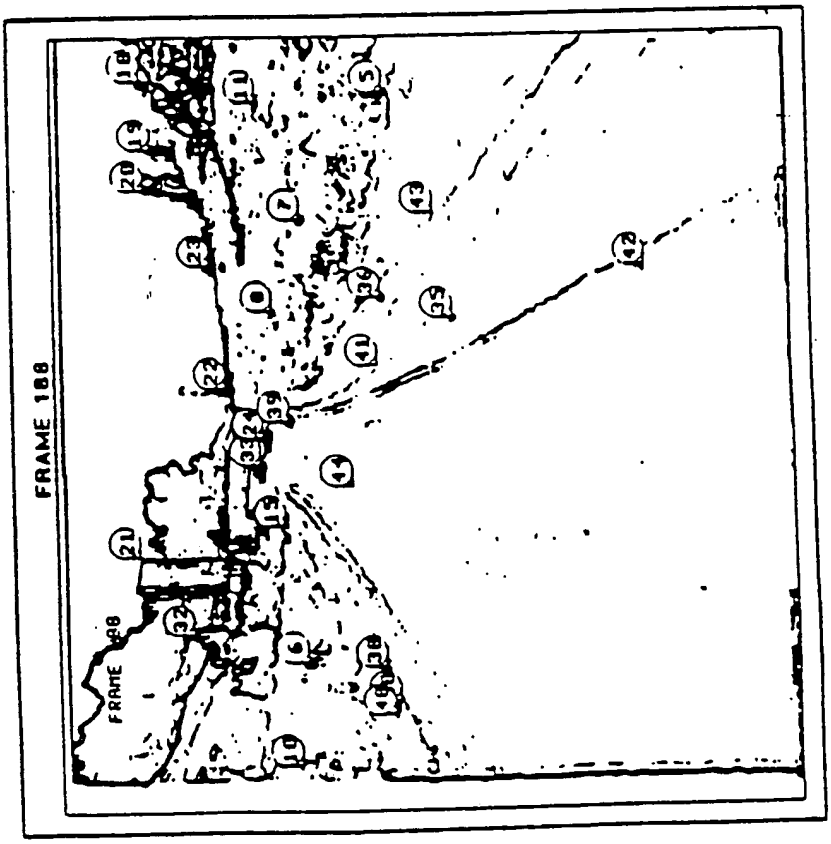


Fig. 35g

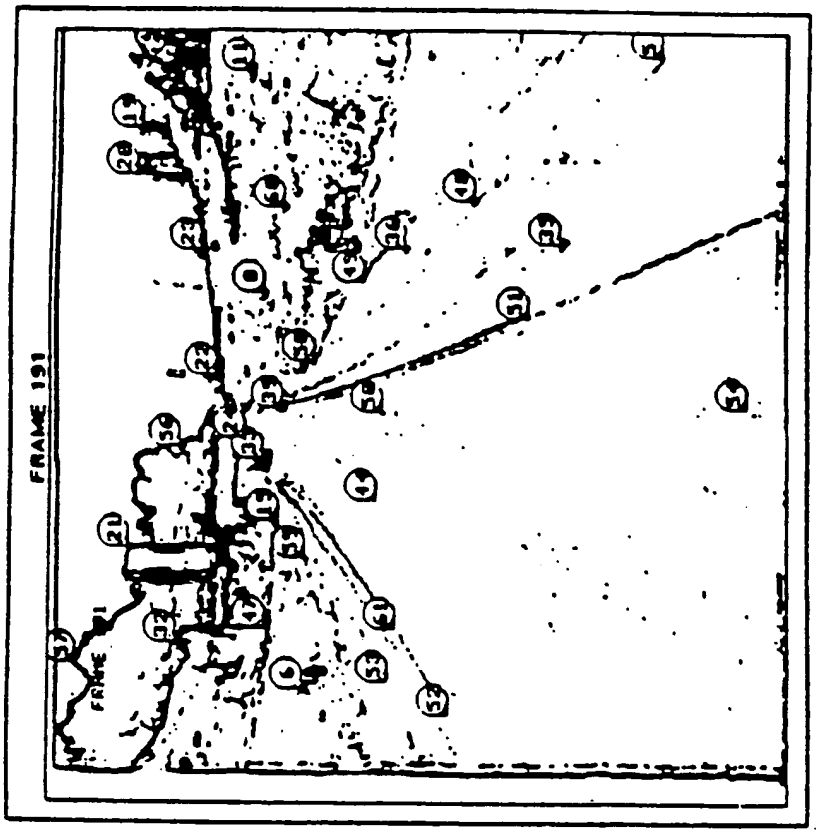


Fig. 35j

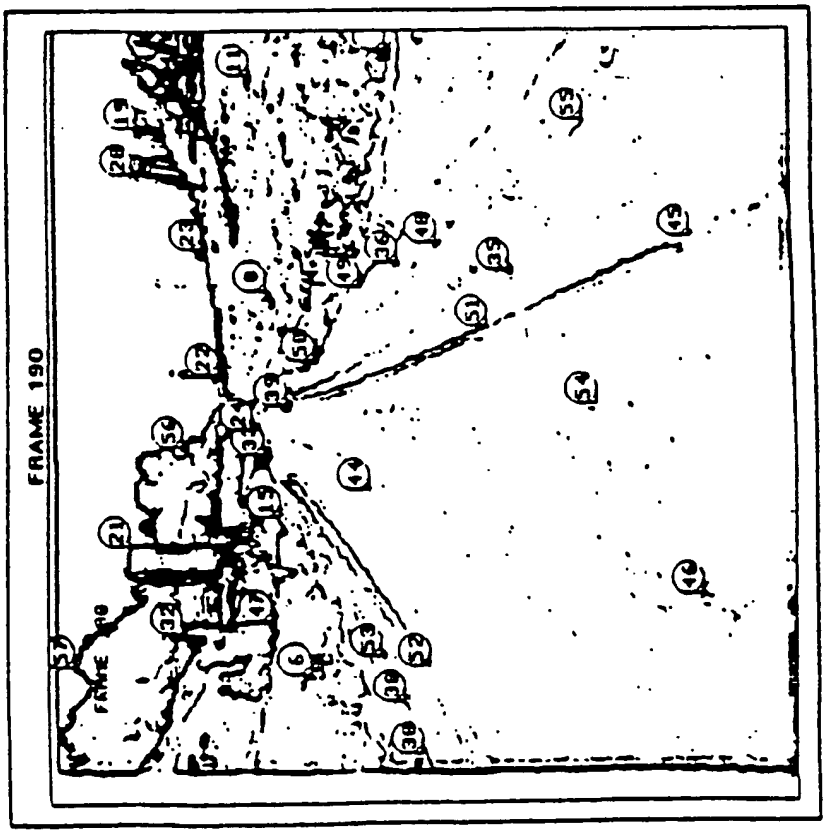


Fig. 35i

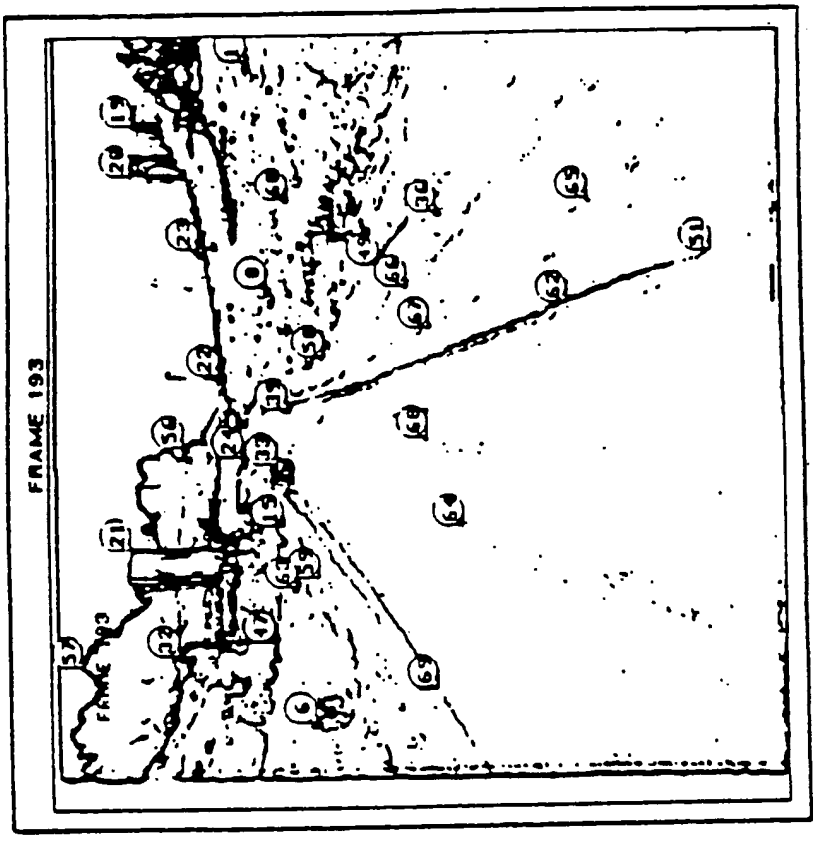


Fig. 35l

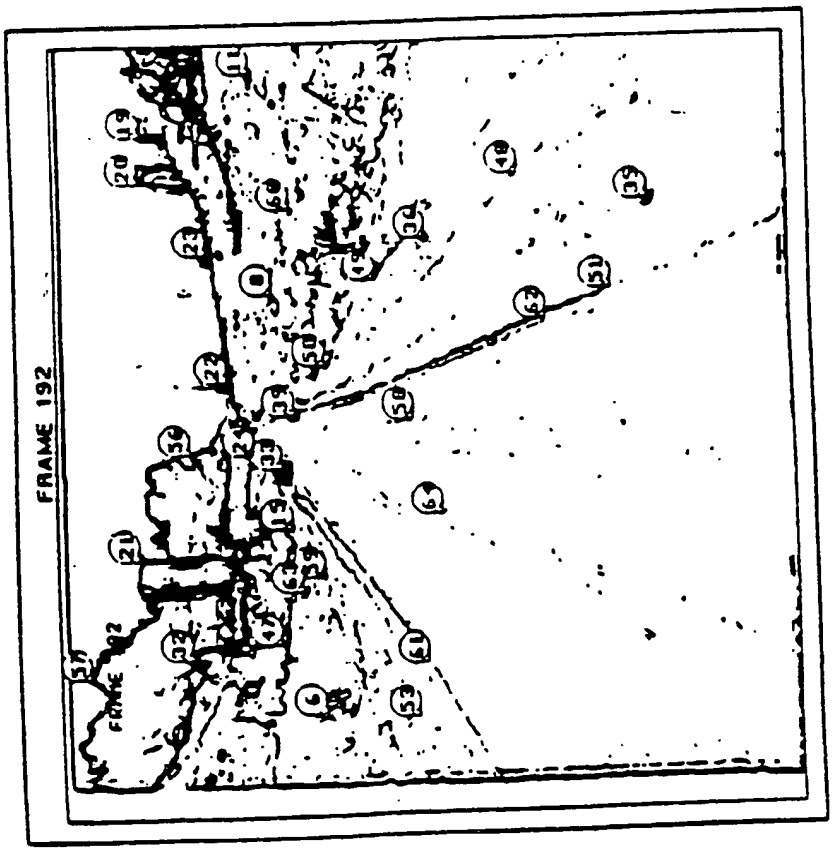


Fig. 35k

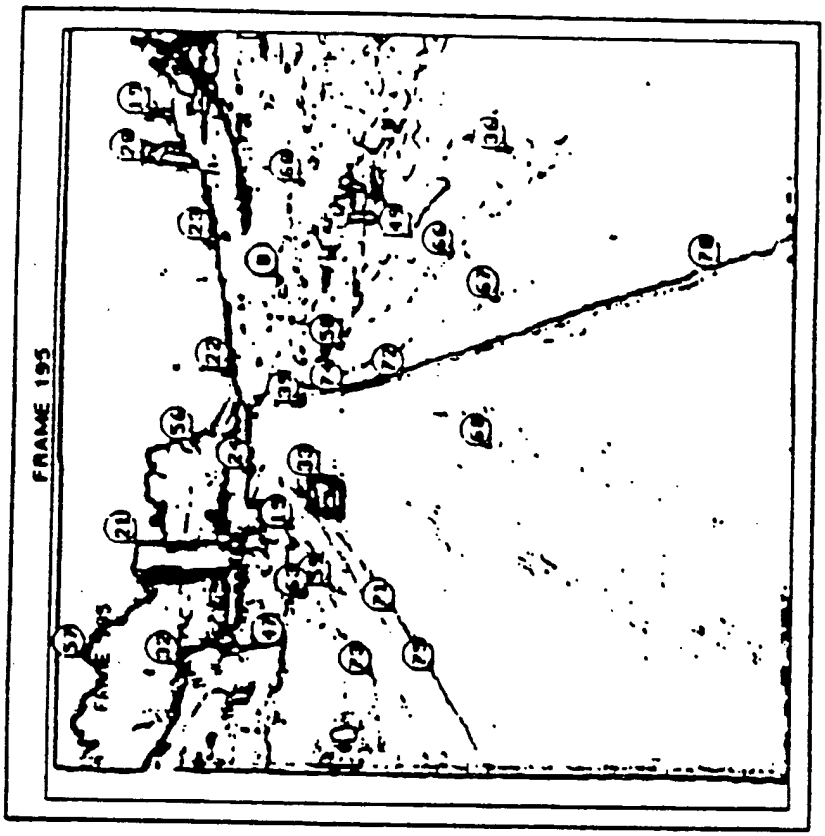


Fig. 35n

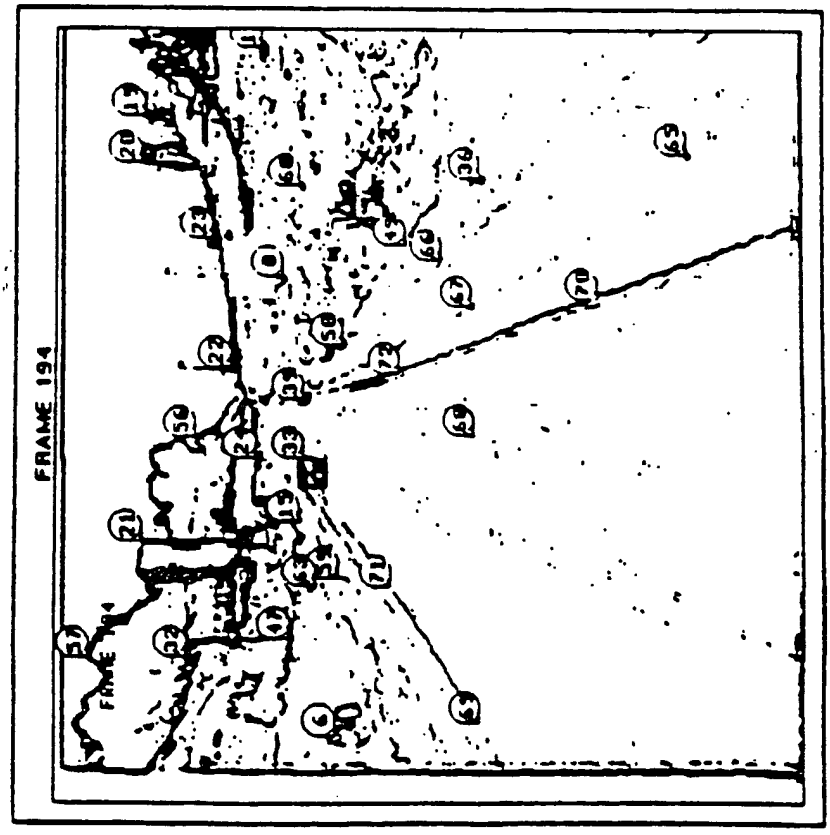


Fig. 35m

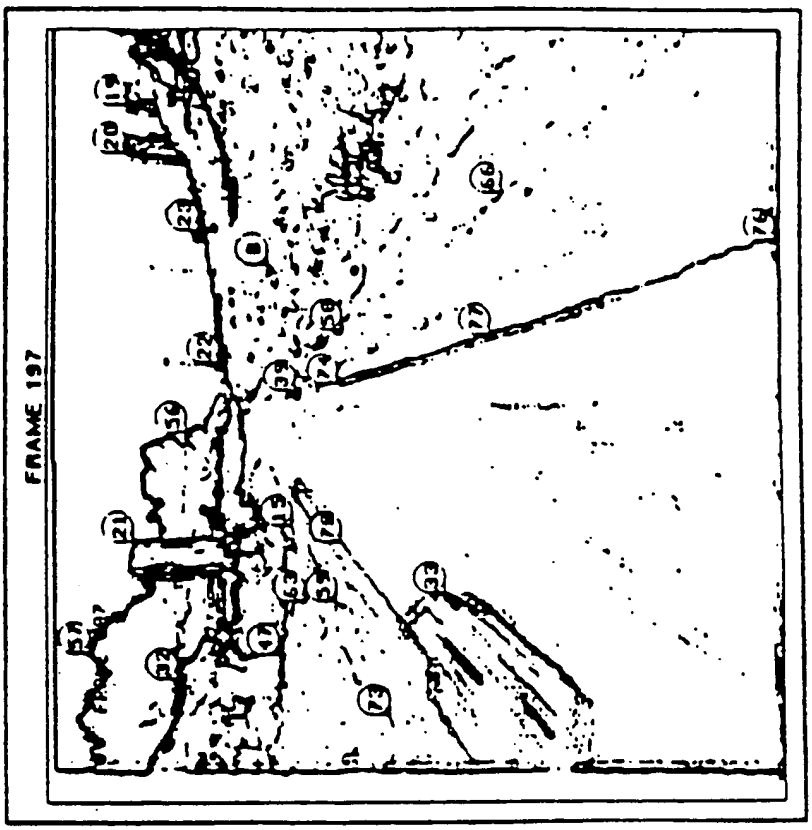


Fig. 35p

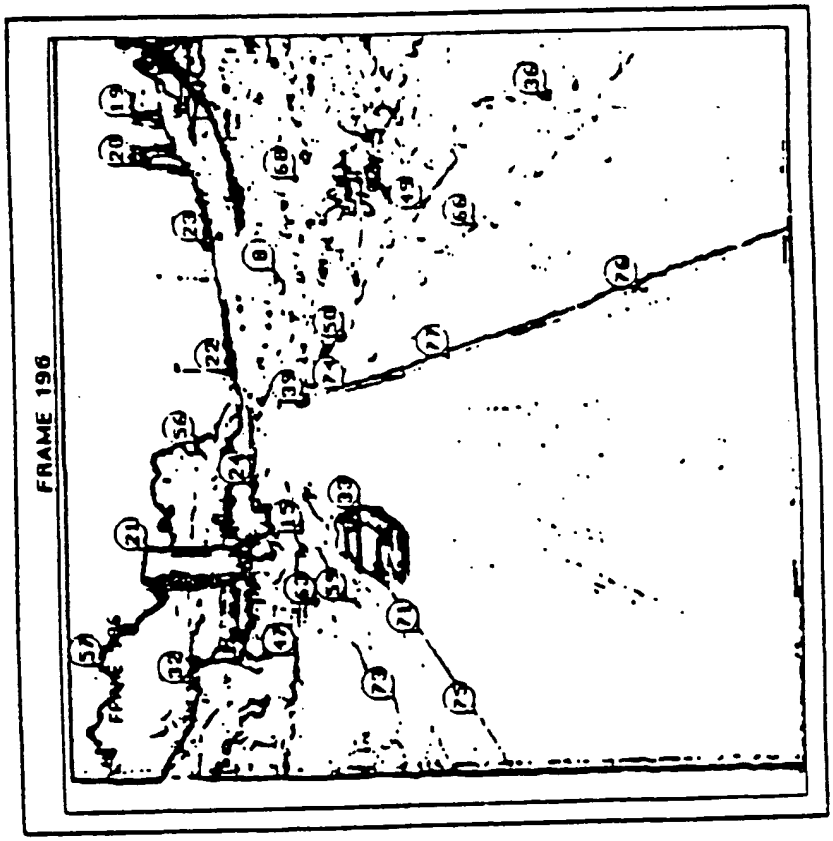


Fig. 35o

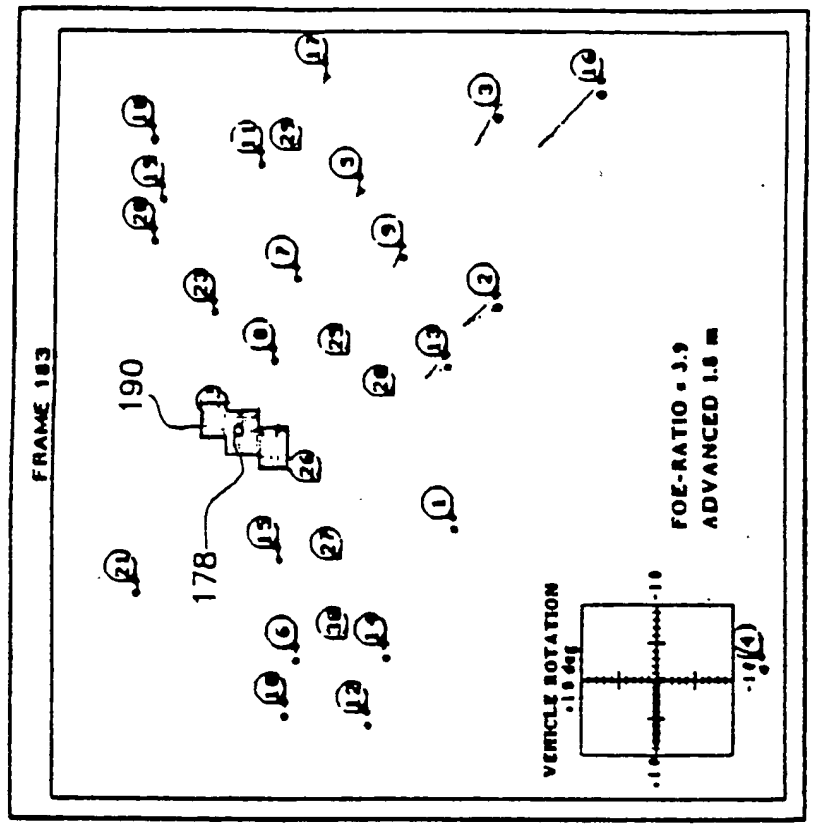


Fig. 36b

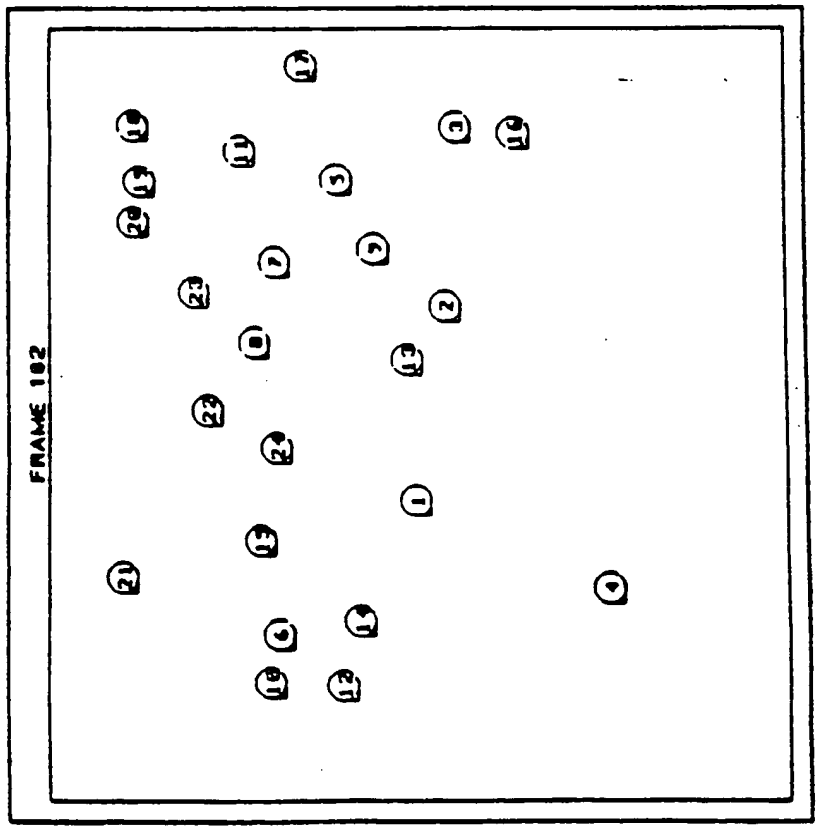


Fig. 36a

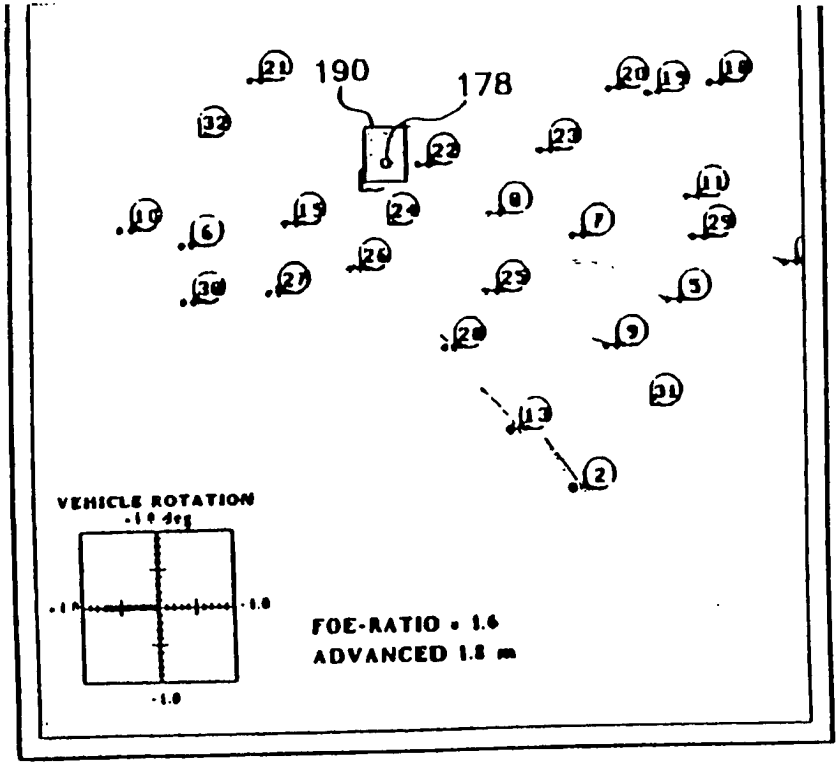


Fig. 36c

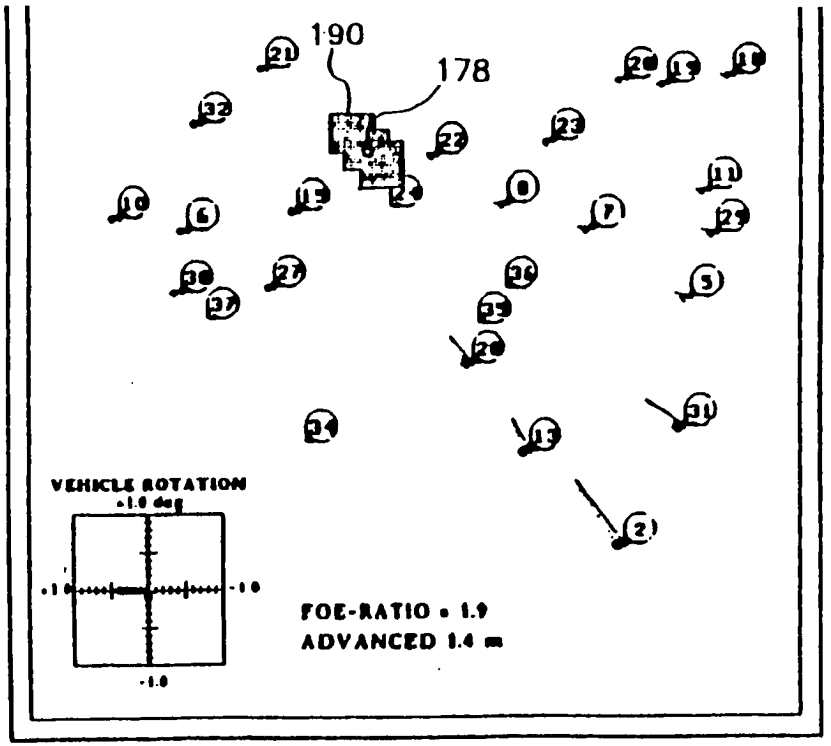


Fig. 36d

EP 0 390 051 A2

A 4 1 1 2 7 3 0

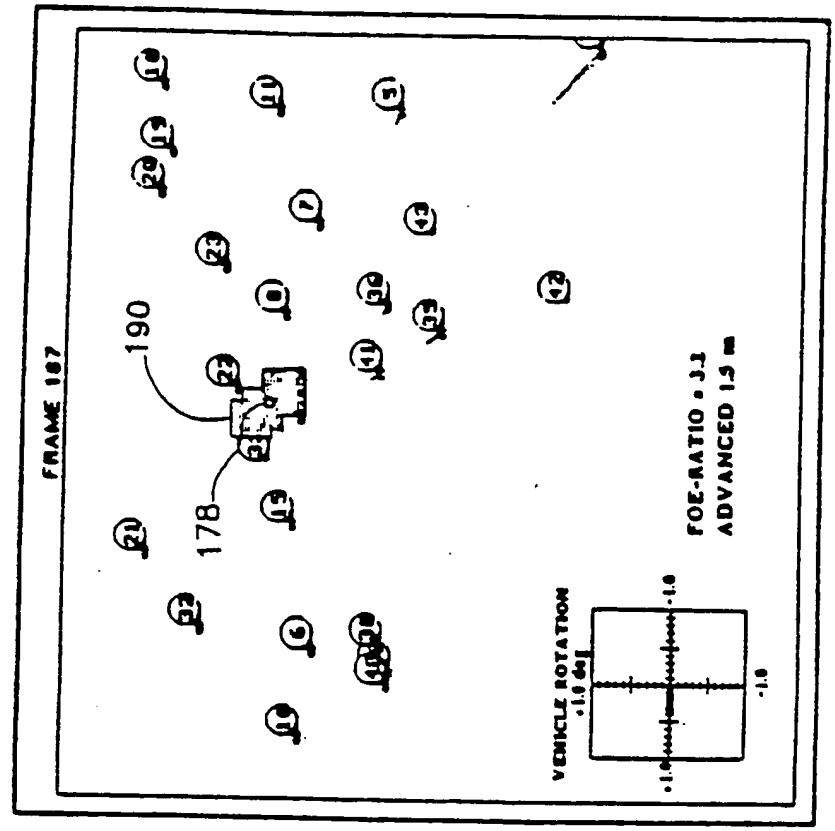


Fig. 36f

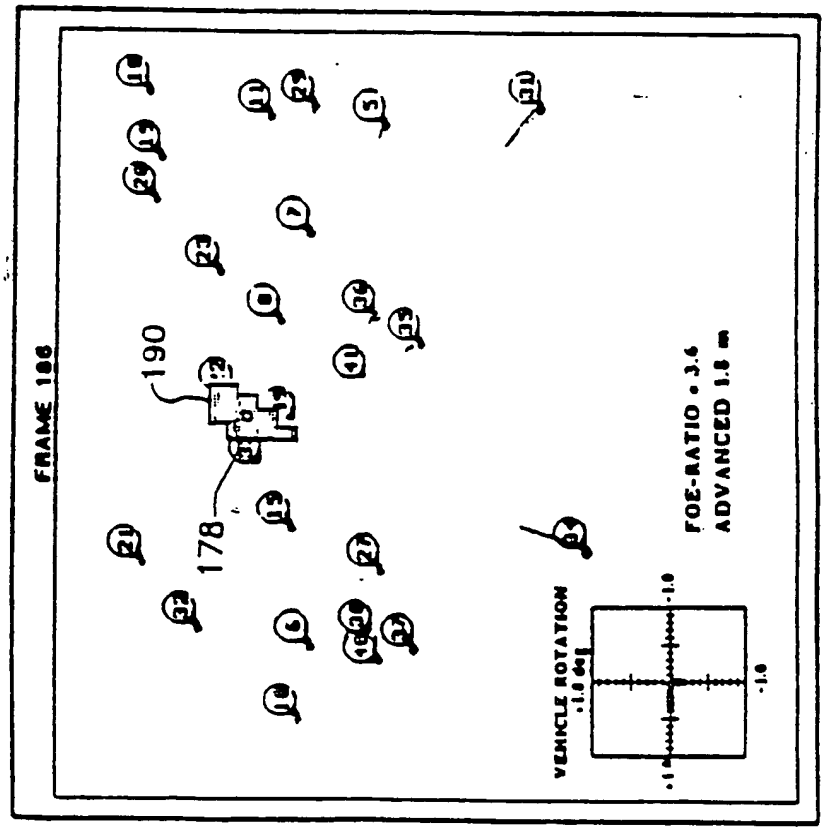


Fig. 36e

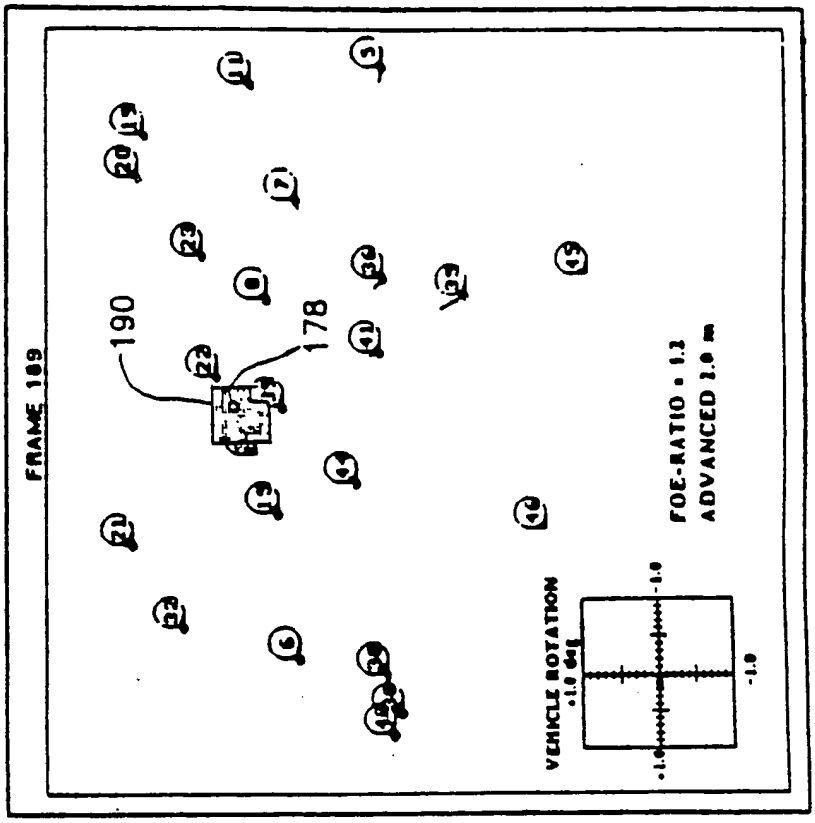


Fig. 36h

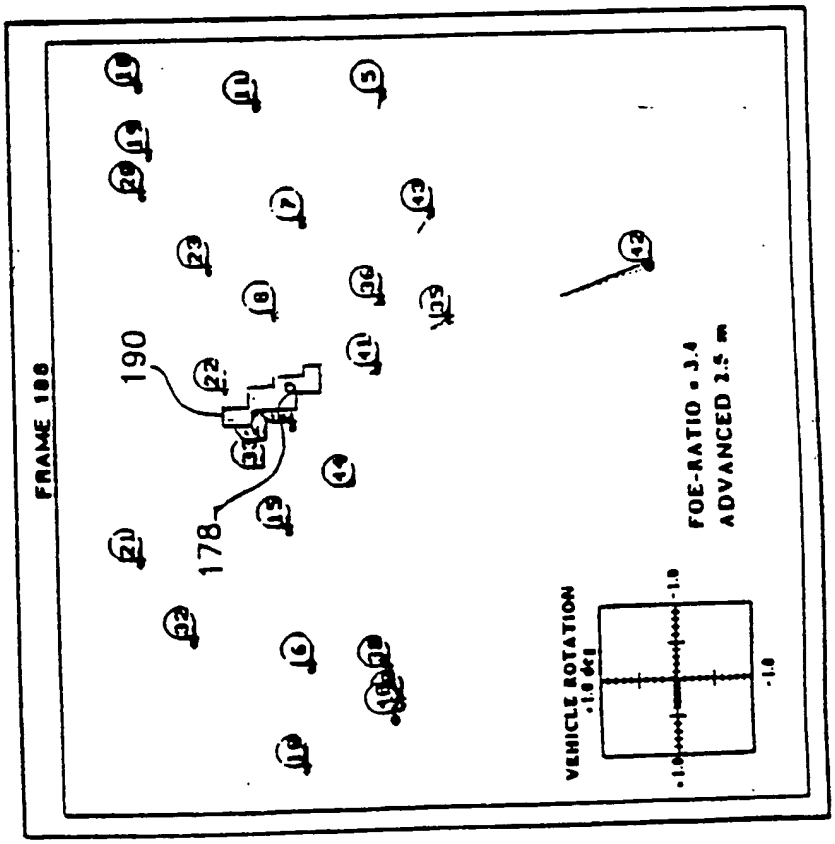


Fig. 36g

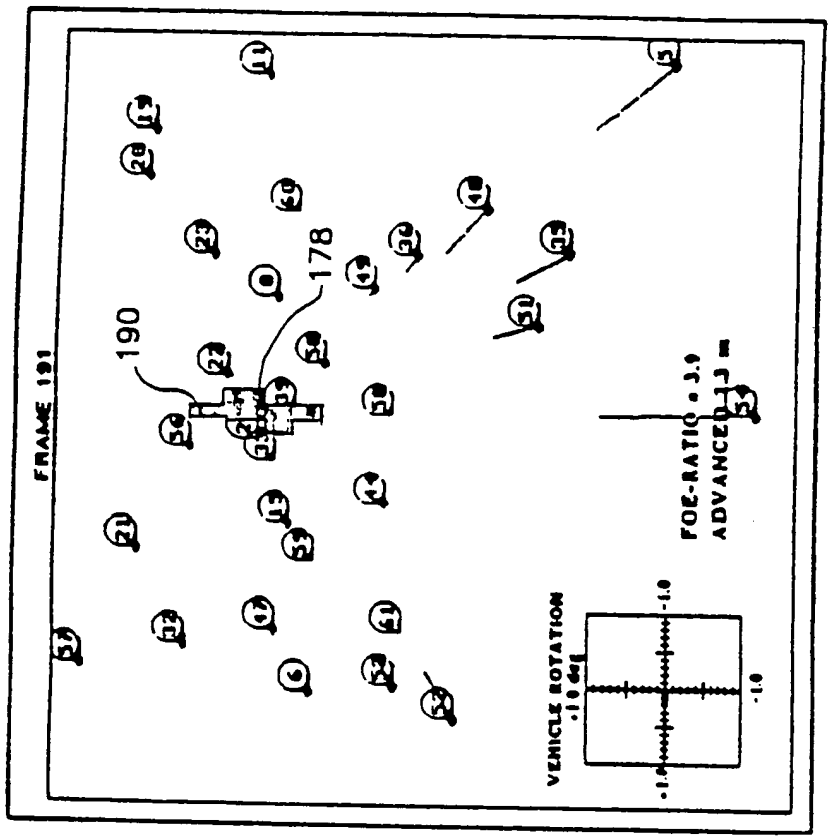


Fig. 36j

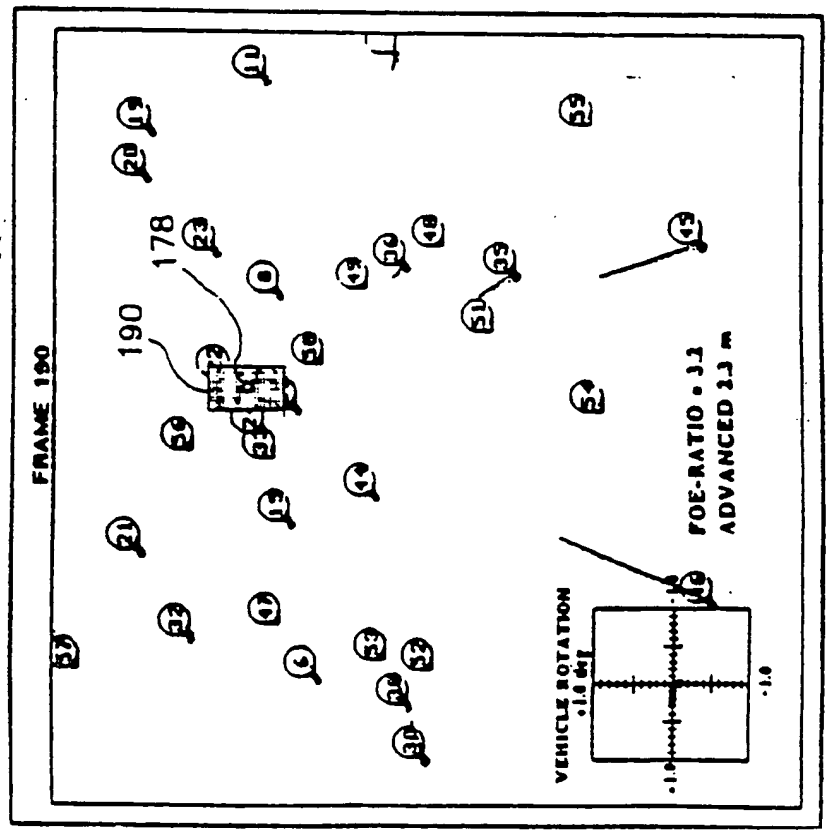


Fig. 36i

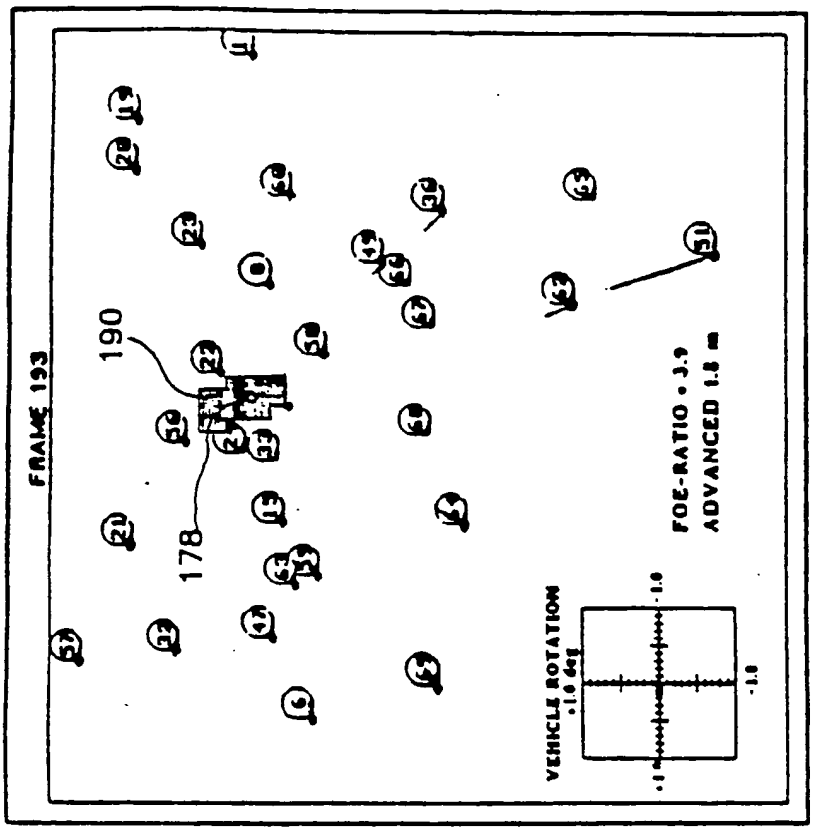


Fig. 36l

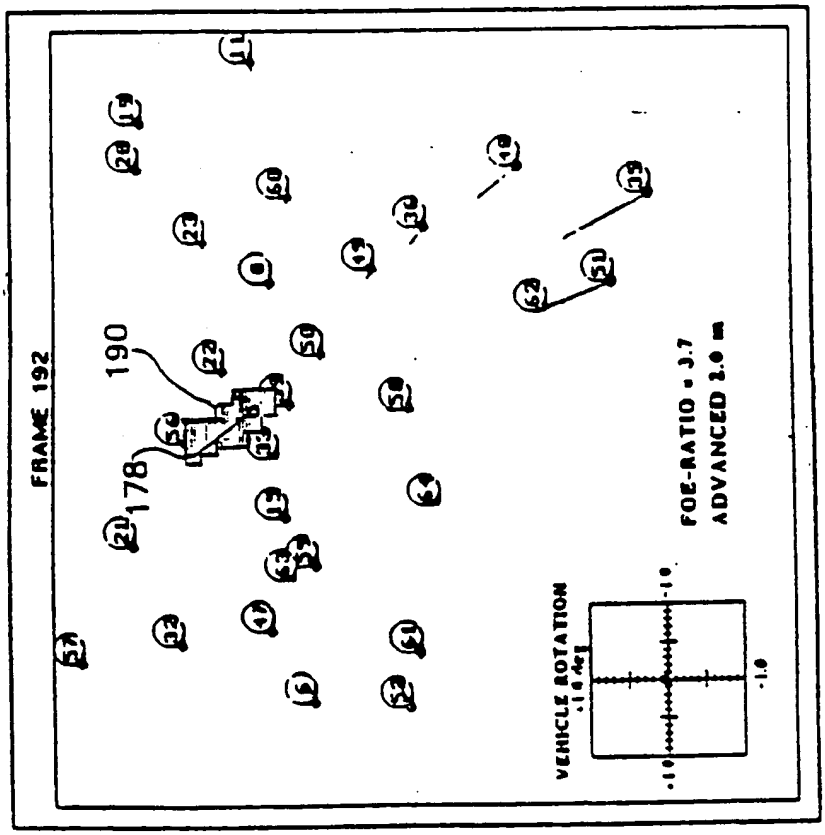


Fig. 36k

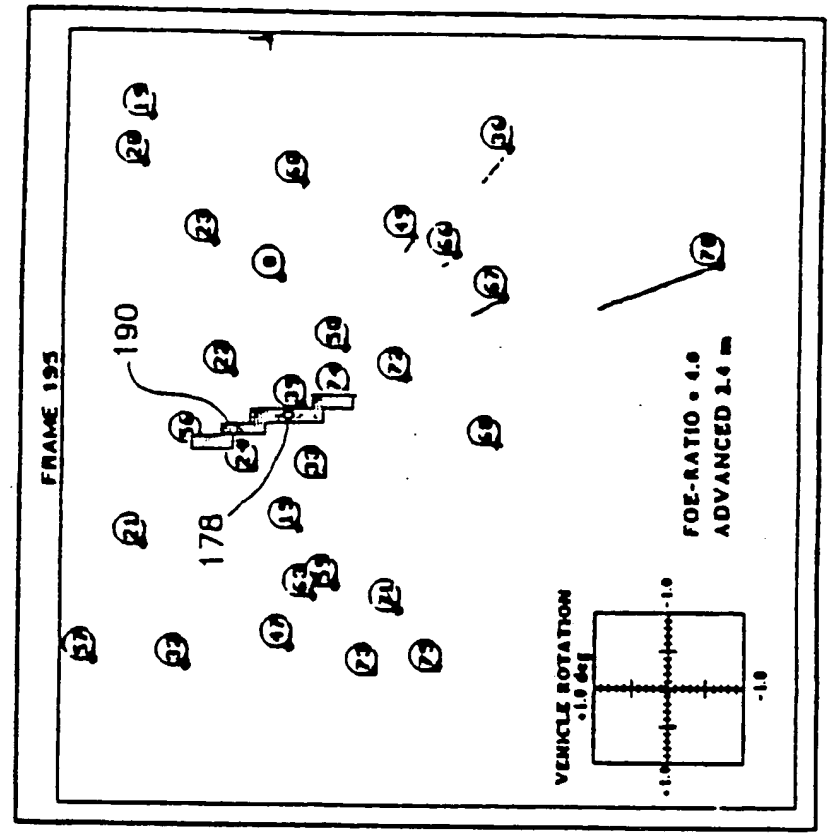


Fig. 36n

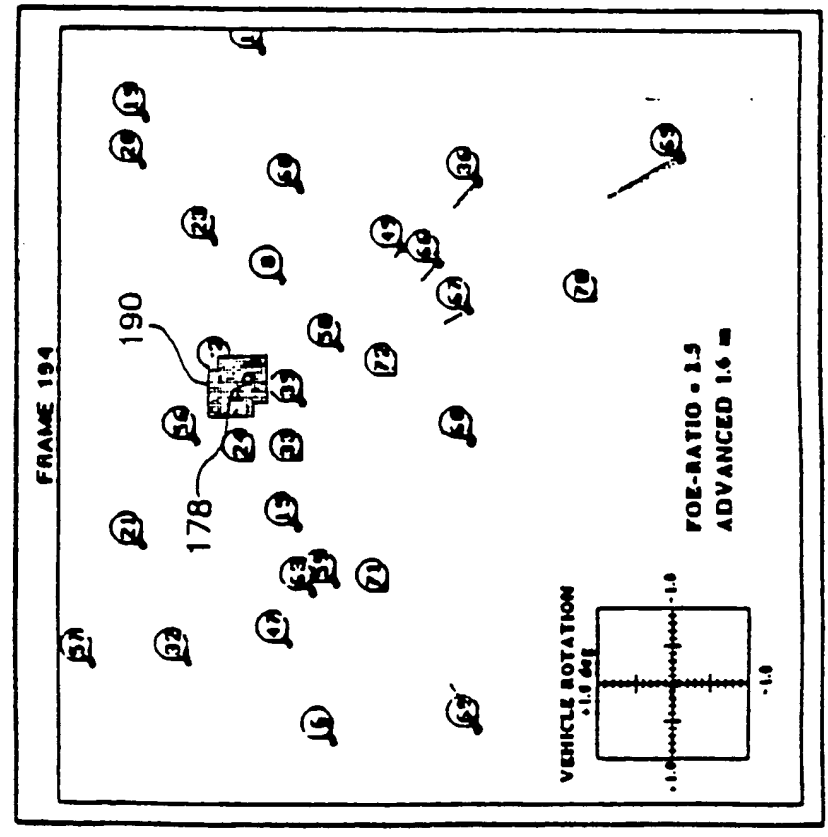


Fig. 36m

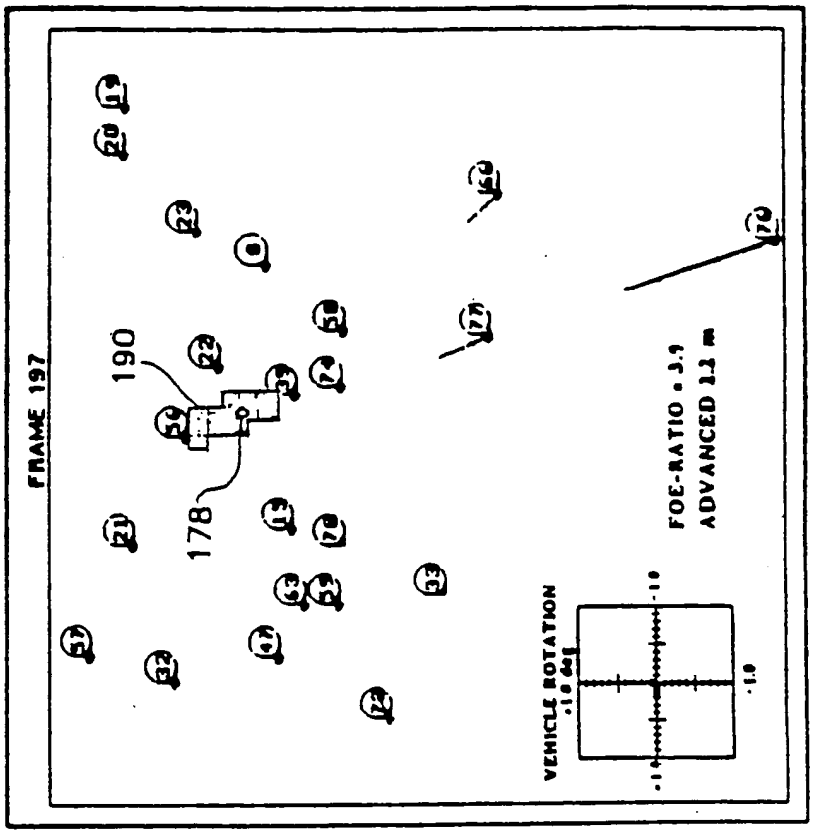


Fig. 36p

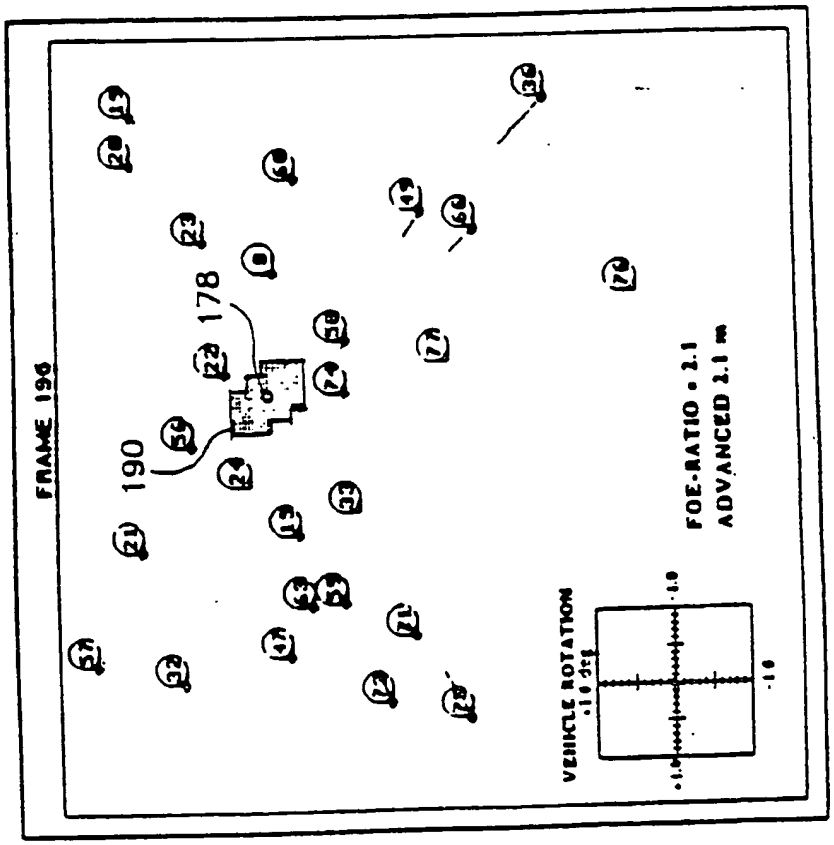


Fig. 36o

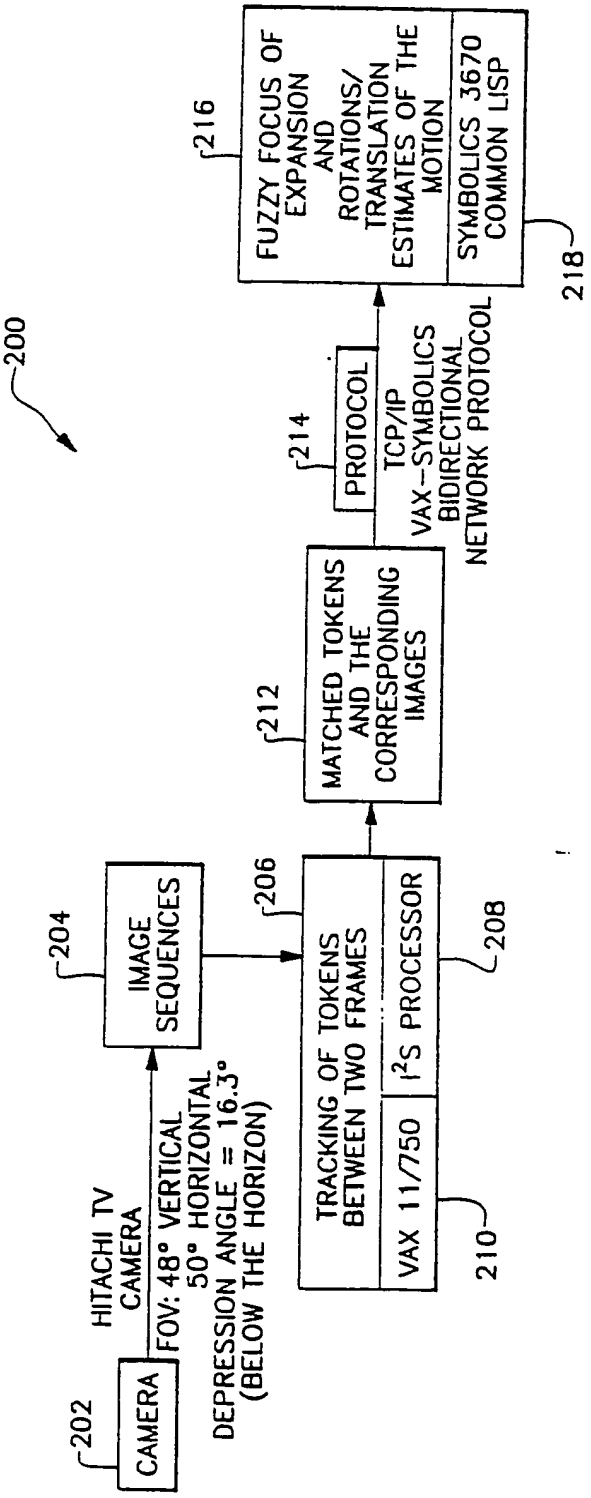

200
202
CAMERA
HITACHI TV CAMERA
FOV: 48° VERTICAL
50° HORIZONTAL
DEPRESSION ANGLE = 16.3°
(BELOW THE HORIZON)
204
IMAGE SEQUENCES
206
TRACKING OF TOKENS BETWEEN TWO FRAMES
VAX 11/750
I²S PROCESSOR
208
210
212
MATCHED TOKENS AND THE CORRESPONDING IMAGES
214
PROTOCOL
TCP/IP
VAX–SYMBOLICS
BIDIRECTIONAL
NETWORK PROTOCOL
216
FUZZY FOCUS OF EXPANSION AND ROTATIONS/ TRANSLATION ESTIMATES OF THE MOTION
SYMBOLICS 3670
COMMON LISP
218

Fig. 37

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 390 051 A3**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number: **90105762.0**

(51) Int. Cl.5: **G05D 1/08**

(22) Date of filing: **27.03.90**

(30) Priority: **31.03.89 US 332320**

(43) Date of publication of application:
03.10.90 Bulletin 90/40

(84) Designated Contracting States:
DE FR GB NL SE

(88) Date of deferred publication of the search report:
31.07.91 Bulletin 91/31

(71) Applicant: **HONEYWELL INC.**
Honeywell Plaza
Minneapolis MN 55408(US)

(72) Inventor: **Bhanu, Bir**
1889 29th N.W.
New Brighton, MN 55112(US)
Inventor: **Burger, Wilhelm**
Hagenstrasse 61/5
A-4040 Linz(AT)

(74) Representative: **Herzbach, Dieter et al**
Honeywell Europe S.A. Holding KG Patent & License Dept. Kaiserleistrasse 55 Postfach 10 08 65
W-6050 Offenbach am Main(DE)

(54) **Method and apparatus for computing the self-motion of moving imaging devices.**

(57) Determining the self-motion in space of an imaging device (e.g., a television camera) by analyzing image sequences obtained through the device. Three-dimensional self-motion is expressed as a combination of rotations (ϕ,θ,ψ) about the horizontal and vertical camera axes and the direction of camera translation. The invention computes the rotational and translational components of the cameras self-motion exclusively from visual information. Robust performance is achieved by determining the direction of heading (i.e., the focus of expansion) as a connected region instead of a single location on the image plane. The method can be used to determine when the direction of heading is outside the current field of view and when there is zero or very small camera translation.

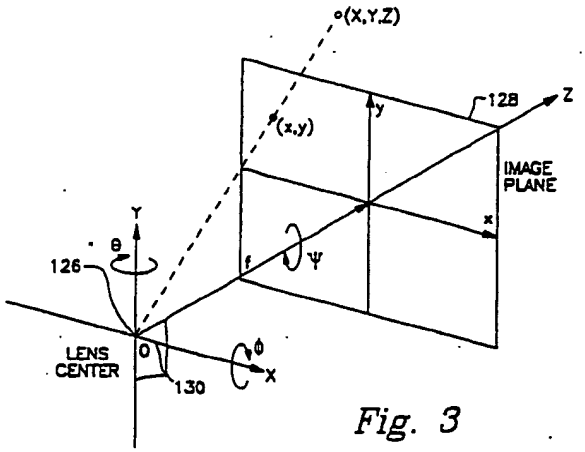


Fig. 3

EP 0 390 051 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number

EP 90105762.0

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
X	DE - A1 - 3 604 401 (MESSERSCHMITT-BÖLKOW-BLOHM) * Fig. 1,2; Abstract; claims *	1-3	G 05 D 1/12 G 05 D 1/08
A	* Totality *	4-8	
X	WO - A1 - 87/02 797 (ENERGY OPTICS) * Fig. 2,5,8; abstract; claims *	3	
A	* Fig. 2,5,8,9; abstract, claims *	1,2,4-7	
A	WO - A1 - 89/02 622 (MESSERSCHMITT-BÖLKOW-BLOHM) * Fig. 1B,4C,4D,6E,7A-C; claims *	1-8	
A	FR - A1 - 2 628 859 (KABUSHIKI TOSHIBA) * Fig. 1-3; abstract *	1-5,7	
			TECHNICAL FIELDS SEARCHED (Int. Cl.5)
A	US - A - 4 357 661 (LAMBREGTS) * Fig. 1-6; abstract; claims *	1,3,7,8	G 05 D 1/00

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
VIENNA	19-03-1991	KRAL

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document